Eplaza - An Online Shopping Assistant

Group Members

Arslan Anwar (01-131182-006)

Muhammad Shahab (01-131182-024)

Supervisor: Dr. Tamim Ahmed Khan

A Final Year Project submitted to the Department of Software Engineering, Faculty of Engineering Sciences, Bahria University, Islamabad in the partial fulfillment for the award of degree in Bachelor of Software Engineering

July 2022

THESIS COMPLETION CERTIFICATE

Student Name: Arslan Anwar Enrolment No: 01-131182-006

Student Name: Mohammad Shahab Enrolment No: 01-131182-024

Programme of Study: Bachelor of Software Engineering

Project Title: Eplaza - An Online Shopping Assistant

It is to certify that the above students' project has been completed to my satisfaction and to my belief, its standard is appropriate for submission for evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at ___14%___ that is within the permissible limit set by the HEC. I have also found the thesis in a format recognized by the department.

Supervisor's Signature: ___________________________

Date: July 4, 2022 Name: Dr. Tamim Ahmed Khan

CERTIFICATE OF ORIGINALITY

This is certified that the intellectual contents of the project **Eplaza - An Online Shopping Assistant** are the product of our own work except, as cited properly and accurately in the acknowledgements and references, the material taken from such sources as research journals, books, internet, etc. solely to support, elaborate, compare, extend and/or implement the earlier work. Further, this work has not been submitted by me/us previously for any degree, nor it shall be submitted by me/us in the future for obtaining any degree from this University, or any other university or institution. The incorrectness of this information, if proved at any stage, shall authorities the University to cancel my/our degree.

Name of the Student: Arslan Anwar

Signature: ______________ Date: July 4, 2022

Name of the Student: Mohammad Shahab

Signature: ______________ Date: July 4, 2022

Abstract

As the world is moving towards digitalization and each business is taking the aid of technology to make their businesses flourish, we’ve developed a system that connects not only the local sellers to their potential customers but also can be used to market their products. This project is a web-based shopping system for local sellers as well as entrepreneurs who need a platform to sell their articles. The goal of the project is to create an online shopping platform that is easy to use from both perspectives, the buyer, and the seller. It allows the customers to buy their favourite items from an already existing credible shop without any fear of getting a wrong or faulty product through our website. It also facilitates the local vendors to create their shops and reach many potential customers. This will assist them to sell more and reach a larger customer base. The shop owners will be running their physical store as well as the website’s orders, so they will not be lagging in terms of the use of the technology. The trend of buying from home is increasing rapidly after the pandemic, the customers are already waiting for credible sources to buy from. Our sellers also have a physical shops and they are using our services for web presence, which adds to their integrity. We have our application running at http://www.eplazapk.com/ and we have M&P courier as the delivery service provider to deliver the purchased products on behalf of online shops and businesses. Customers can track orders, discounts and coupon features, customer level system and social media integration. Our system is an effort to take businesses of our country to the information age ready for internationalization with businesses able to go worldwide.

Dedication

To our parents for their love and support

Acknowledgments

First and foremost, we would like to thank Allah Almighty for the continuous showering of blessing that He has bestowed upon us throughout our lives.

We would like to thank Dr. Tamim Ahmed Khan, our supervisor. This FYP would not have been possible without his assistance and active participation in every step of the process. We would like to express my gratitude for your patience and understanding over the last few months.

Getting through our dissertation required more than just academic assistance, and we are grateful to so many people for listening to and, at times, tolerating us over the last three years. We can't begin to express how grateful we are for their friendship. Throughout our time at the University, Habib-Ur-Rehman and Mohammad Umar Farooq provided unwavering personal and professional support.

Lastly, none of this could have happened without our families. Their love and continuous support helped us throughout the progress of the ePlaza.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

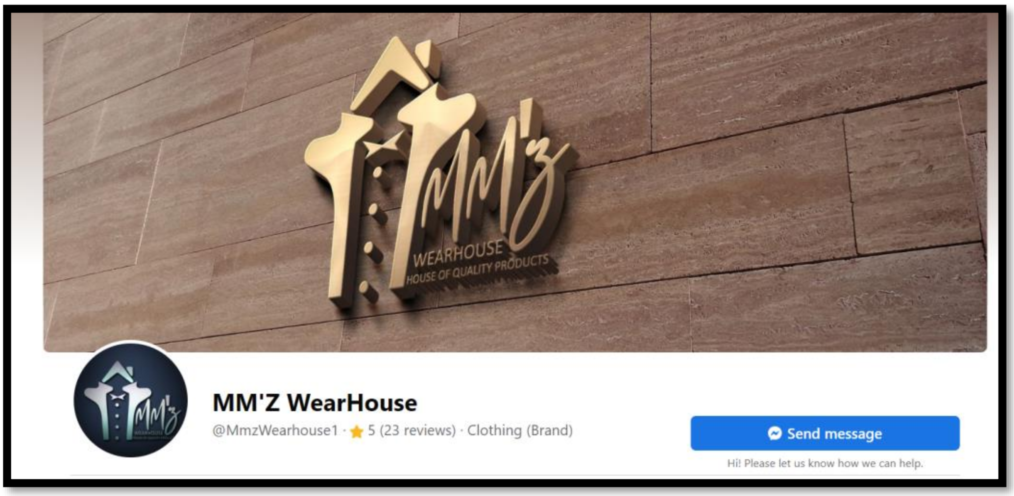
INTRODUCTION

Introduction

This chapter discusses the system's origins as well as the basic idea or purpose that inspired us to create ePlaza and how its contribution can benefit both us and society.

1.1. Motivation

The goal of developing an ePlaza is to establish our own e-commerce platform that serves both local and social sellers. The success of Pakistan’s existing ecommerce platforms is one of the motivating elements for us. Secondly, one of the motives stems from our experience creating an internet business called MMz Wearhouse., as shown below



MM'Z WearHouse

@MmzWearhouse1 · ★ 5 (23 reviews) · Clothing (Brand)

Hi! Please let us know how we can help.

Figure 2:1 MMz Facebook Page

By gaining experience from our previous online venture, we learned about business ethics as well as the technical feasibility of an online business. But the main problem was that we were unable to scale that business due to a lack of automation. We changed the technical path and developed a robust framework that will help us to atomize orders.

Thirdly, the pandemic had made businesses shift their focus from physical to online medium. Which has paved the way for websites like **eplazapk.com**.

1.2. Problem Statement

Existing e-commerce platforms do not assist local dealer's businessmen who are working around the clock to make ends meet. Their platform is only for educated individuals who have learned how to handle businesses, among other things. During a social crisis, such as a pandemic, a medium is needed to aid local, social, and home sellers, people who sell on social media sites such as factbook have less options to grow their businesses.

1.3. Objectives

The primary goal of ePlaza is to satisfy both sellers (those who create shops and add products) and customers (those who buy items from shops) in terms of system utilisation and services provided by ePlaza.

- For sellers, ePlaza strives to provide a platform for local shopkeepers or wholesale dealers to sell items.
- On the other side, consumers can visit any shop, buy things, interact with interactive screens, share their ideas, and many more possibilities are available to please them.

Three main objectives are

1. A platform for Consumer & Seller.
2. Compelling features to stick Consumers on our system.
3. Social Media Integration to support social selling.

1.4. Main contributions

1.4.1. New features

ePlaza allows its sellers to automate Facebook page posts in terms of creating, editing, and deleting posts; this is one of the best ways of product marketing and engaging people from all over the country to visit our system and buy products. In addition, a Level System is being implemented to keep customers motivated to order daily. Customers can take advantage of exciting offers at each level up.

1.4.2. Beneficial for society

Through the Covid epidemic, we learned about the uncertainties in the offline business, such as how all local shop owners were struggling to keep their businesses open during the lockdown. As a result, creating the ePlaza platform will be a source of solace for them in times of crisis. They can quickly transition to an online platform at no expense.

1.4.3. System Recognition

Free marketing, social selling support, online store management, customer care, and product photography are the initial free services we provide to get our system known.

1.5. Report organisation

Chapter 1 introduction of the document, systems motivation, goal, and its importance in society.

Chapter 2 brief overview of system background and throwing light in comparison of existing and our system.

Chapter 3 focuses on system requirements in detail.

Chapter 4 discuss the design of the system, covering structure as well architecture of system.

Chapter 5 discusses the implementation and development of system about sort of technology we are using and approaches.

Chapter 6 consists of discussion related to the testing of the system.

Chapter 7 concludes the thesis by summarizing different aspects of the work. Contributions of the current work and further developments to the work are also proposed in this chapter.

Chapter 2

BACKGROUND STUDY

2.1. Background

Our System background study is based on our previous experiences in the ecommerce field, where we encountered shortcomings in scaling up our business, eventually leading to its closure. They were as follows:

- Market Saturation: Similar products at low cost are available.
- Inadequate marketing experience.
- Product originality
- A lack of a proper platform, such as a website, to scale our business.

We began targeting customers through the android platform before shutting down by launching the Android app MMZ.

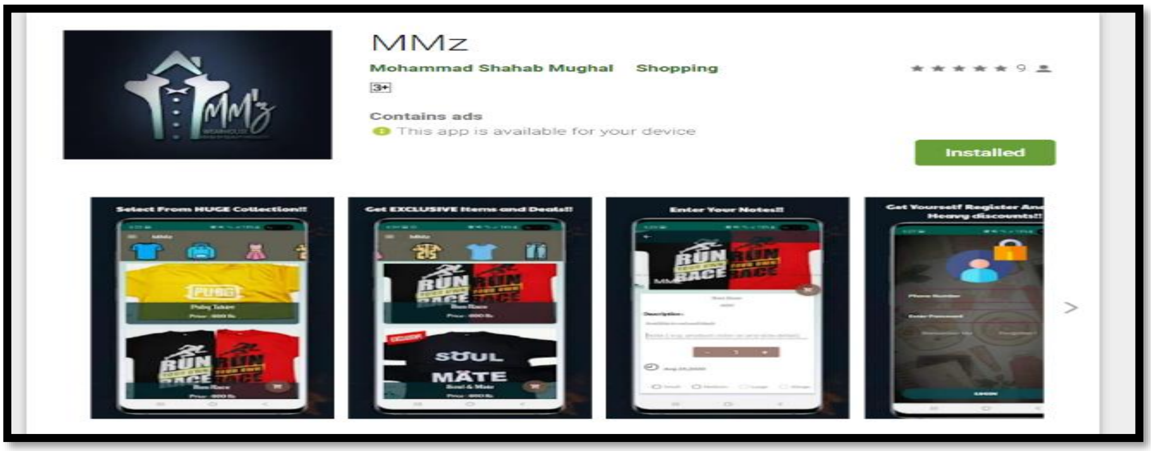

MMz
Mohammad Shahab Mughal
Shopping
3+
Contains ads
This app is available for your device
9
Installed
Select From HUGE Collection!!
Get EXCLUSIVE Items and Deals!!
Enter Your Notes!!
Get Yourself Register And Heavy discounts!!
SOUL MATE
RUN RACE
PUBG

Figure 2:1.1 MMz App

The mobile app failed, forcing us to conclude that to maintain a position in such a field, a website is required where, instead of selling products on our own, local sellers can publish their products and give us a cut for our services from using the platform to drop shipping. System serves as a location where multiple vendors can set up shop and sell their goods, giving customers more options.

2.2. Related Ecommerce Websites

2.2.1. Daraz

Daraz has been an e-commerce pioneer since its debut in 2012. We've grown up with ever-increasing daraz. It was an inspiration, and we did our hardest to imitate Daraz's purchasing mechanism. And we've created a framework for local enterprises to thrive.

The following are some of the important aspects from Daraz that inspired us and that we improved.

Category Tree:

Like Daraz tree format category, we also incorporated such system from database schemas. But instead of dropdown covering the entire screen of website as shown in figure 2.2, we introduced this inside a drawer that also supports responsiveness as well.

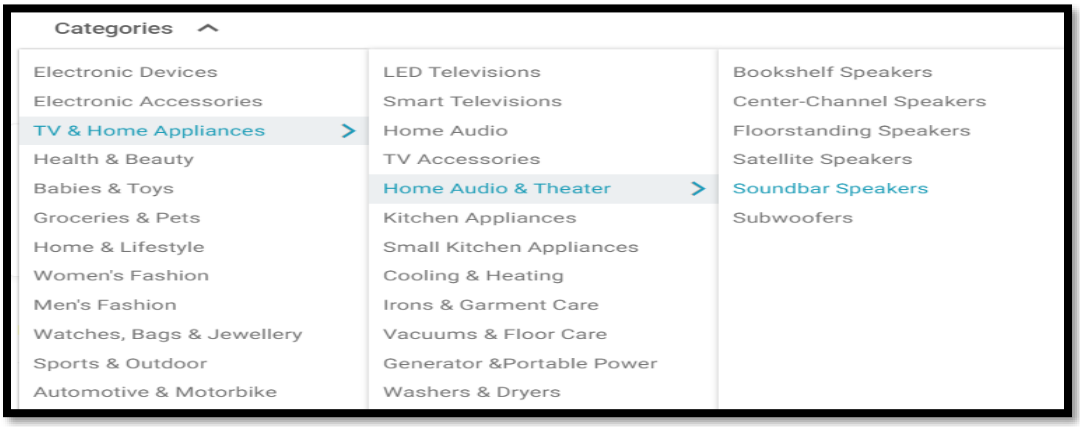


Figure 2.2 Daraz Category Structure

Add to cart checks and logic:

As we can see, the first step in adding an item to our cart is to log in, and then we can add; however, there is a registration option available. Such a process is also satisfied by our system.

Cart Mechanism:

In Daraz every person has its own Cart, and it requires login to access it, however if customer place item in cart and return after a year with different PC or mobile that item would be still there. Same thing is done in our system too.

Pagination:

Daraz using infinite pagination mechanism which considered to be bad as all items are to place on frontend which require more space in frontend to hold such large data. However, we adapted more of classic pagination to our system which simply load specific amount of data to screen.

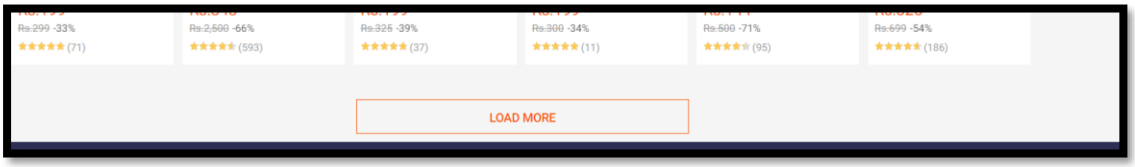


Figure 2.2:2 Daraz Pagination

Coupon Codes:

Like Daraz we also adopted coupon system for our valuable customer, So, that they are attracted to visit our Shop. Not only that we also introduced Level System to publish exciting offer to customers based on his/her Level.

2.2.2. Shopify

Shopify is one of the most popular Ecommerce platforms today, allowing branded stores to be established with their own custom domain. We were inspired by Shopify and decided to establish a gateway for sellers with simple registration requirements. Our portal will allow vendors to open their own shop and sell products, among other things.

However, the Shopify landing page directs us to the subscription screen, but we drive our customers directly to shops and their products, making it easier for them to acquire what they want and preventing any confusion.

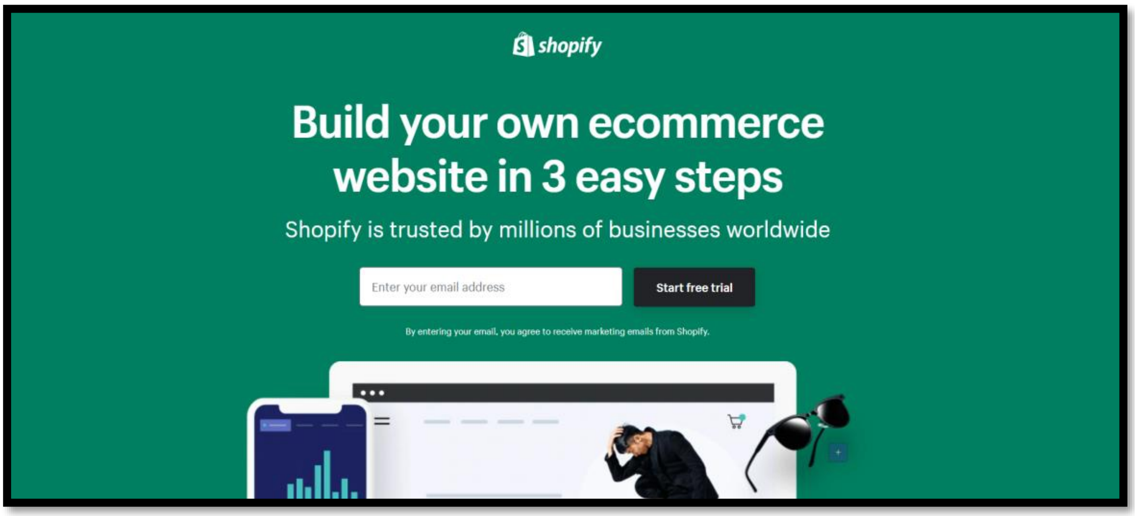
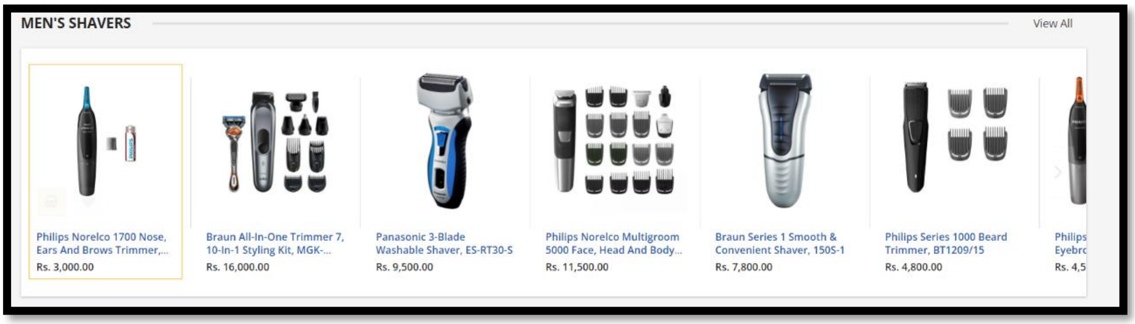

shopify
Build your own ecommerce website in 3 easy steps
Shopify is trusted by millions of businesses worldwide
Enter your email address
Start free trial
By entering your email, you agree to receive marketing emails from Shopify.

Figure 2.2:3 Shoppify

2.2.3. Existing system features

During development, we had our first shop, which was a cosmetic shop, so for a better understanding, we researched one of Pakistan's cosmetic online shops, "Naheed.pk." The following are some of Naheed's characteristics that inspired us.

Home Screen Category Carousels



MEN'S SHAVERS

View All

Philips Norelco 1700 Nose, Ears And Brows Trimmer,...
Rs. 3,000.00

Braun All-In-One Trimmer 7, 10-In-1 Styling Kit, MGK-...
Rs. 16,000.00

Panasonic 3-Blade Washable Shaver, ES-RT30-S
Rs. 9,500.00

Philips Norelco Multigroom 5000 Face, Head And Body...
Rs. 11,500.00

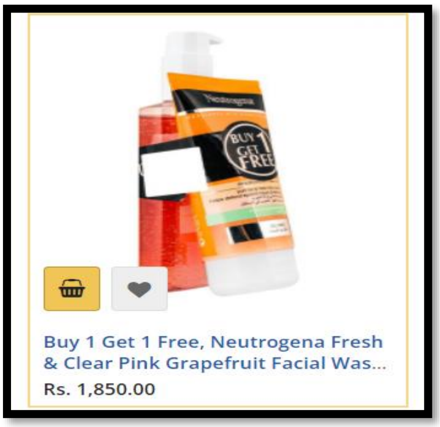
Braun Series 1 Smooth & Convenient Shaver, 150S-1
Rs. 7,800.00

Philips Series 1000 Beard Trimmer, BT1209/15
Rs. 4,800.00

Philips Eyebrc
Rs. 4,5

Figure 2.2:4 Naheed's Category Carousal

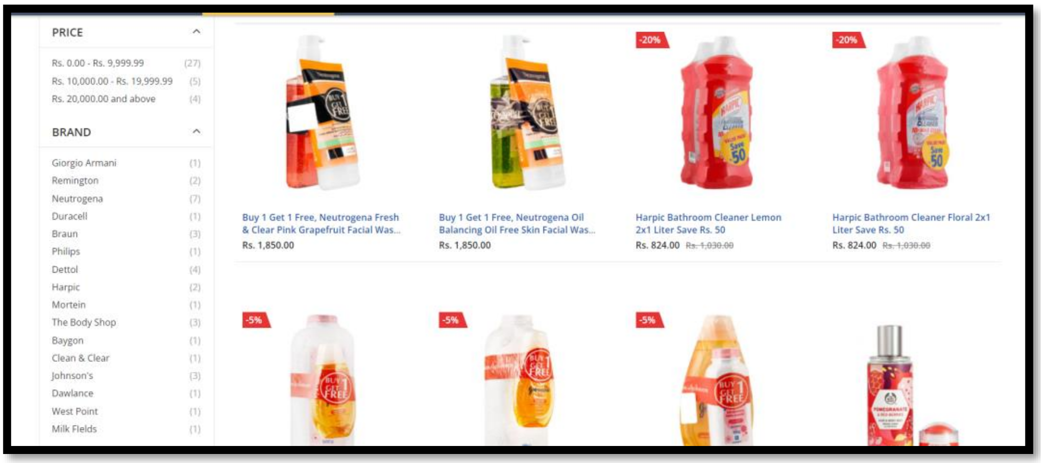
Card



Buy 1 Get 1 Free, Neutrogena Fresh & Clear Pink Grapefruit Facial Was...
Rs. 1,850.00

Figure 2.2:5 Naheed's Card

Search & Filter screen



PRICE	
Rs. 0.00 - Rs. 9,999.99	(27)
Rs. 10,000.00 - Rs. 19,999.99	(5)
Rs. 20,000.00 and above	(4)

BRAND	
Giorgio Armani	(1)
Remington	(2)
Neutrogena	(7)
Duracell	(1)
Braun	(3)
Philips	(1)
Dettol	(4)
Harpic	(2)
Mortein	(1)
The Body Shop	(3)
Baygon	(1)
Clean & Clear	(1)
Johnson's	(3)
Dawlance	(1)
West Point	(1)
Milk Fields	(1)

Buy 1 Get 1 Free, Neutrogena Fresh & Clear Pink Grapefruit Facial Was...
Rs. 1,850.00

Buy 1 Get 1 Free, Neutrogena Oil Balancing Oil Free Skin Facial Was...
Rs. 1,850.00

-20%
Harpic Bathroom Cleaner Lemon 2x1 Liter Save Rs. 50
Rs. 824.00 ~~Rs. 1,030.00~~

-20%
Harpic Bathroom Cleaner Floral 2x1 Liter Save Rs. 50
Rs. 824.00 ~~Rs. 1,030.00~~

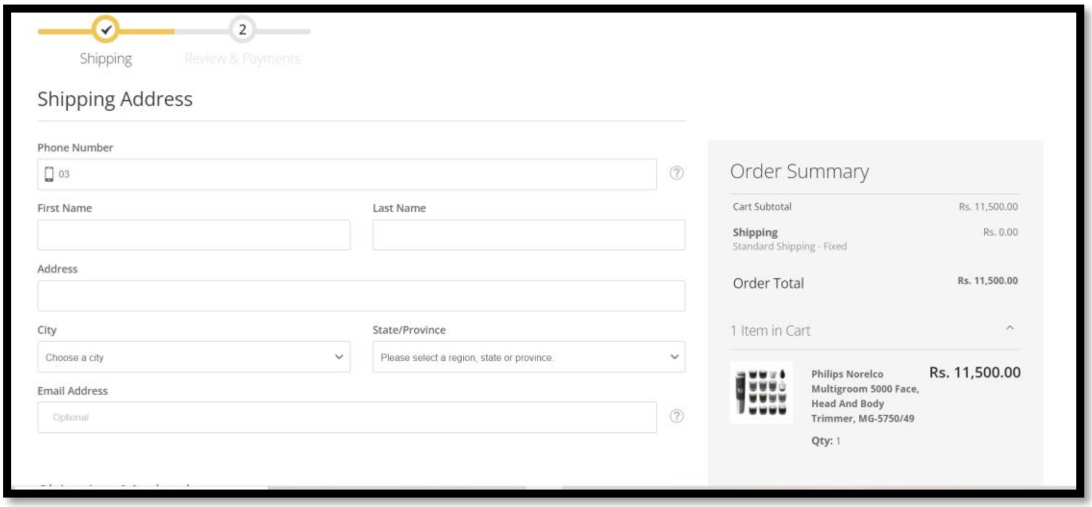
-5%

-5%

-5%

Figure 2.2:6 Naheed's Search Filter Screen

Shipping & Order Summary



Shipping Address

Phone Number
03

First Name

Last Name

Address

City
Choose a city

State/Province
Please select a region, state or province.

Email Address
Optional

Order Summary

Cart Subtotal	Rs. 11,500.00
Shipping Standard Shipping - Fixed	Rs. 0.00
Order Total	Rs. 11,500.00

1 Item in Cart

Philips Norelco Multigroom 5000 Face, Head And Body Trimmer, MG-5750/49
Rs. 11,500.00
Qty: 1

Figure 2.2:7 Naheed's Shipping & Order Summary

Order Tracking System

Just like Naheed we also integrated Order Tracking mechanism through M&P API calls, Customer are free to track their order any time with their order id.

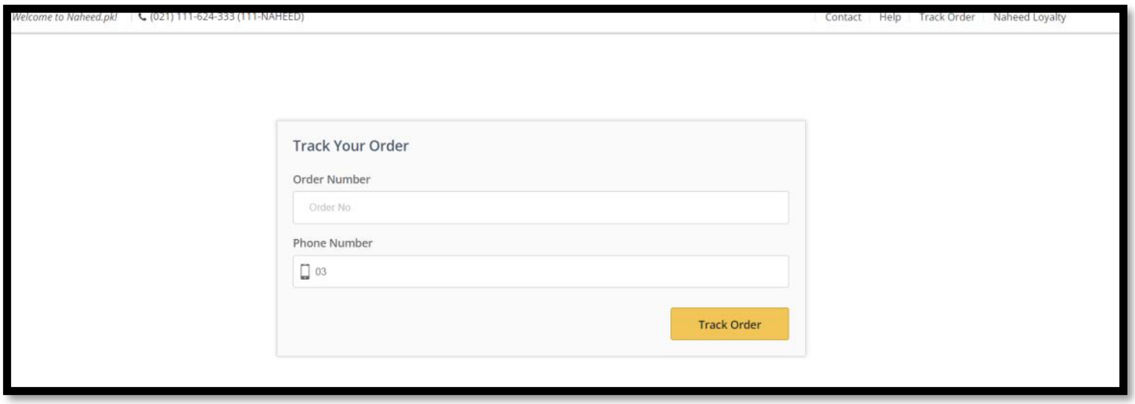


Figure 2.2:8 Naheed's Order Tracking

2.3. Identifying business framework

The tactics for expanding the business and increasing profits are generally based on previous experiences.

- Acquiring items in bulk.
- Customer had not much options to explore and buy.
- Inadequate courier Service

So, based on our previous experience, we devised the following tactics.

- Instead of buying in bulk, we began purchasing items on demand.
- A variety of categories are accessible.
- Being facilitated by multiple vendors.
- After conducting research, we selected M&P courier service as our COD partner; Daraz also employs M&P courier service.

2.4. Identifying technical framework

In terms of technology and platform selection, building an online system became a problem for us. We started looking at what kind of technologies or tools we could use to make the ePlaza framework more powerful. The following questions were derived:

- Which frontend framework should be used?
- Which framework should be utilized for the backend?

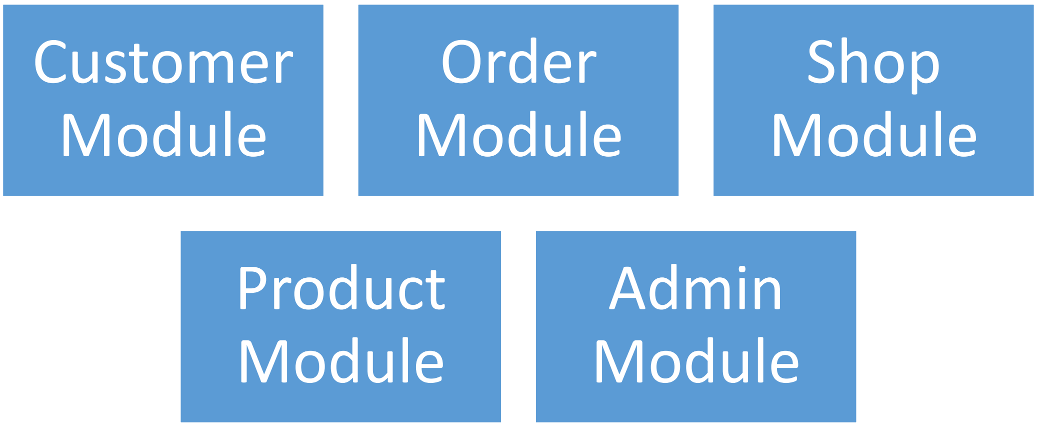
- Should we use a SQL or non-SQL database for our system?
- What kind of authentication measures will be implemented?
- Where should the system be deployed?
- Where will the images be stored?

The queries above aided us in discovering the system from a technical standpoint. However, after extensive research and analysis by our classmates and supervisor, we arrived to following findings.

- React was a good choice for the frontend.
- Django was a good choice for the backend because we had previous familiarity with it.
- A SQL database will be utilized in the system for relational tables.
- Django appears to have built-in authentication features.
- It is agreed that the deployment will take place on the Heroku hosting server.
- Photos to be stored in Amazon Web Services buckets.

2.5. System Key Concepts

The system key Concept can be understood through its modules. These modules comprise the system's fundamental features and design upon which the ePlaza is built. The modules listed below are essential components of ePlaza.



Customer Module

Order Module

Shop Module

Product Module

Admin Module

Figure 2.2:9 System's Modules

2.5.1. Customer

All customer related functionalities are incorporated in this section involves following functionalities

- Customer Login
- Customer Register
- Manage Customer Cart
- Manage Customer Orders
- Get Customer Details
- Manage Customer Level

2.5.2. Order

All Order related functionalities are incorporated in this section involves following functionalities

- Get Order List based on Shops
- Placing Order
- Generating emails on Placing order
- Updating order Status
- Track Order

2.5.3. Shop

All shop related functionalities are incorporated in this section involves following functionalities

- Seller Login Register
- Get Shop Details
- Shop Register
- Shop Stats
- Shop management

2.5.4. Product

All Product related functionalities are incorporated in this section involves following functionalities

- Get Products on search & filtration
- Get Products by shop

- Review product
- Store, update, delete product
- Social media posting and updating via Graph API
- And many more.

2.5.5. Admin

All Admin related functionalities are incorporated in this section involves following functionalities

- Manage Categories
- Home screen Carousels
- Shop Verification
- Stats

Chapter 3

SYSTEM DESIGN

3.1. Use Case Diagrams

3.1.1. Seller Use Case Diagram

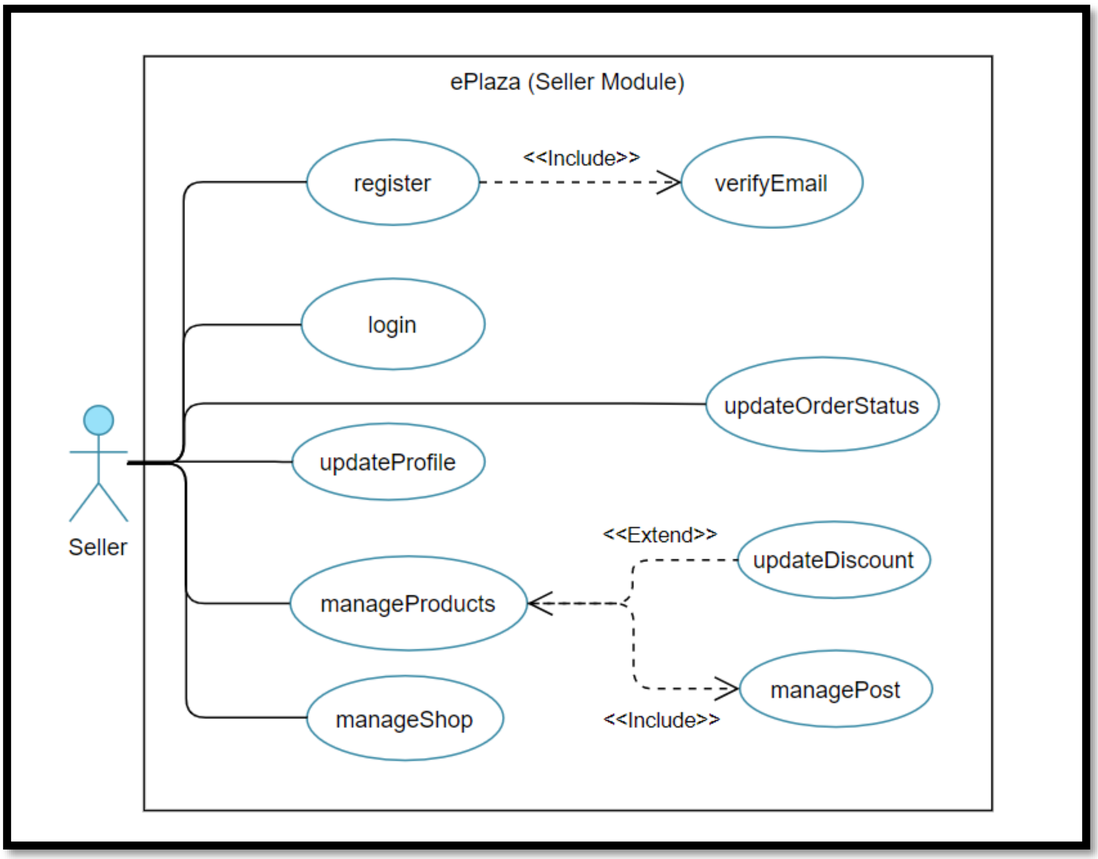


Figure 3:1 Seller User Case Diagram

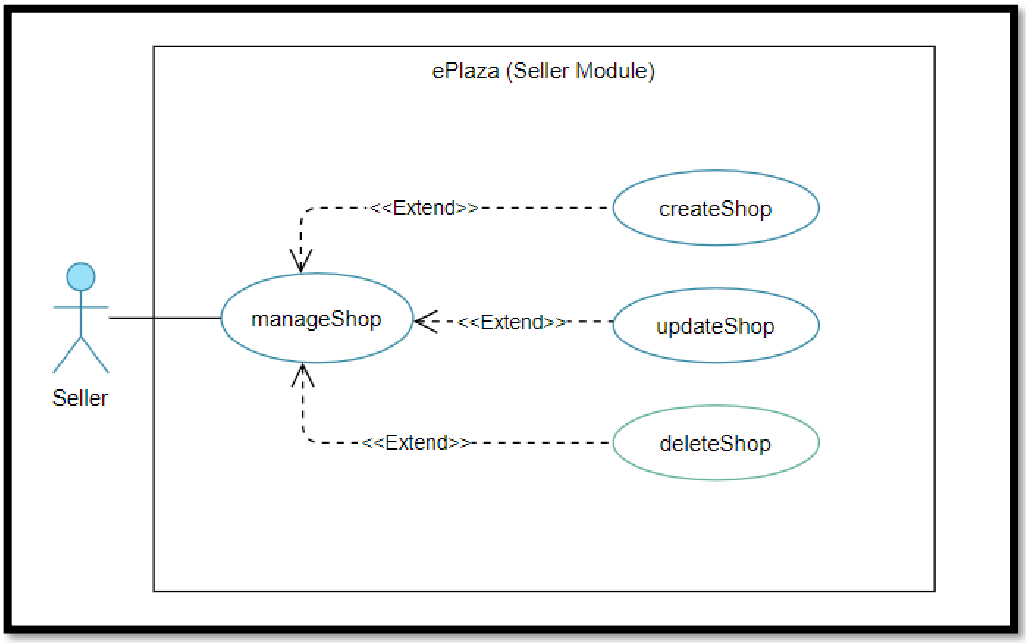


Figure 3:2 Seller Use Case Diagram Manage Shop

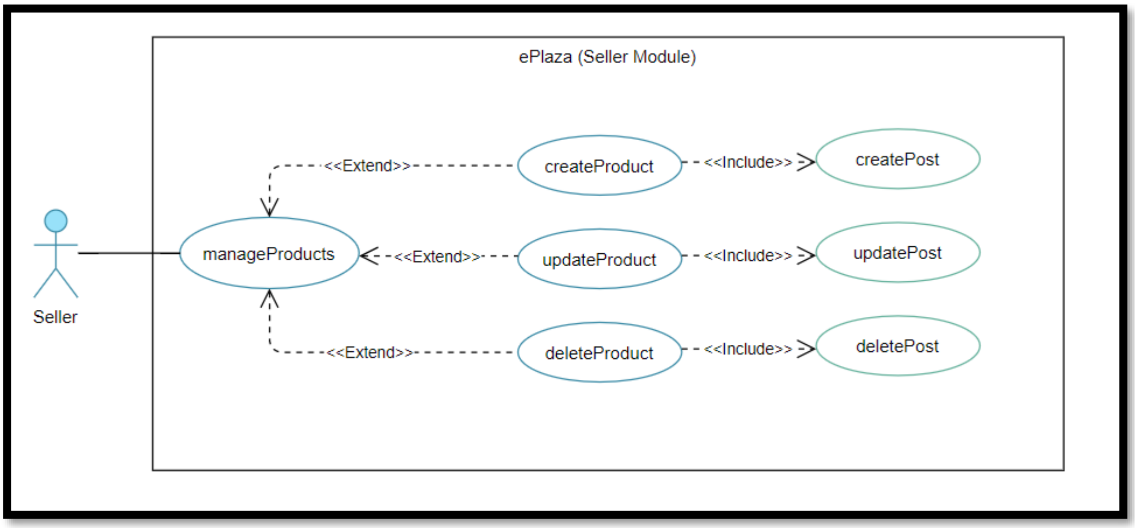


Figure 3:3 Seller Use Case Diagram Manage Products

3.1.2. Customer Use case Diagram

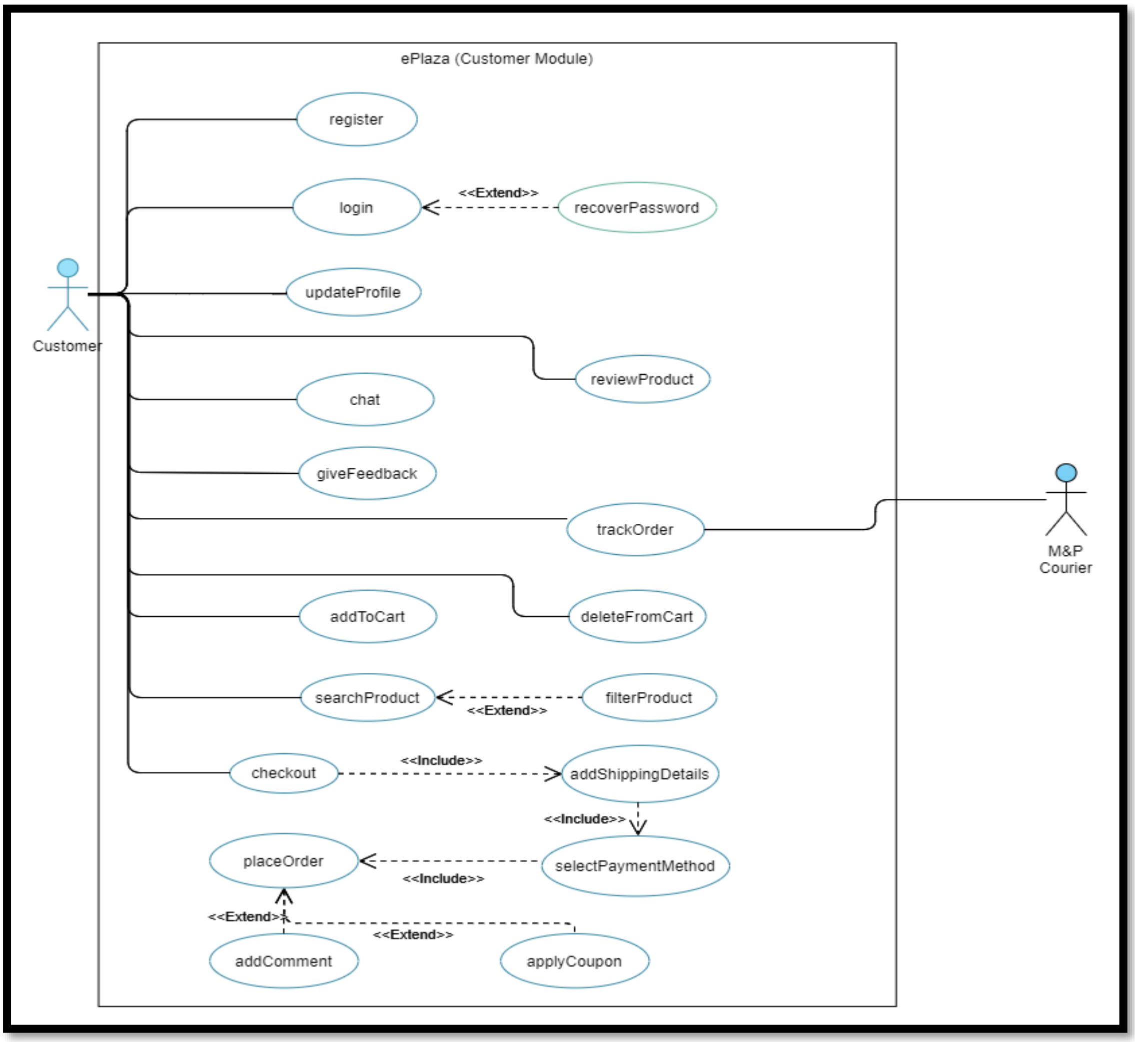


Figure 3:4 Customer Use Case Diagram

3.1.3. Admin Use case Diagram

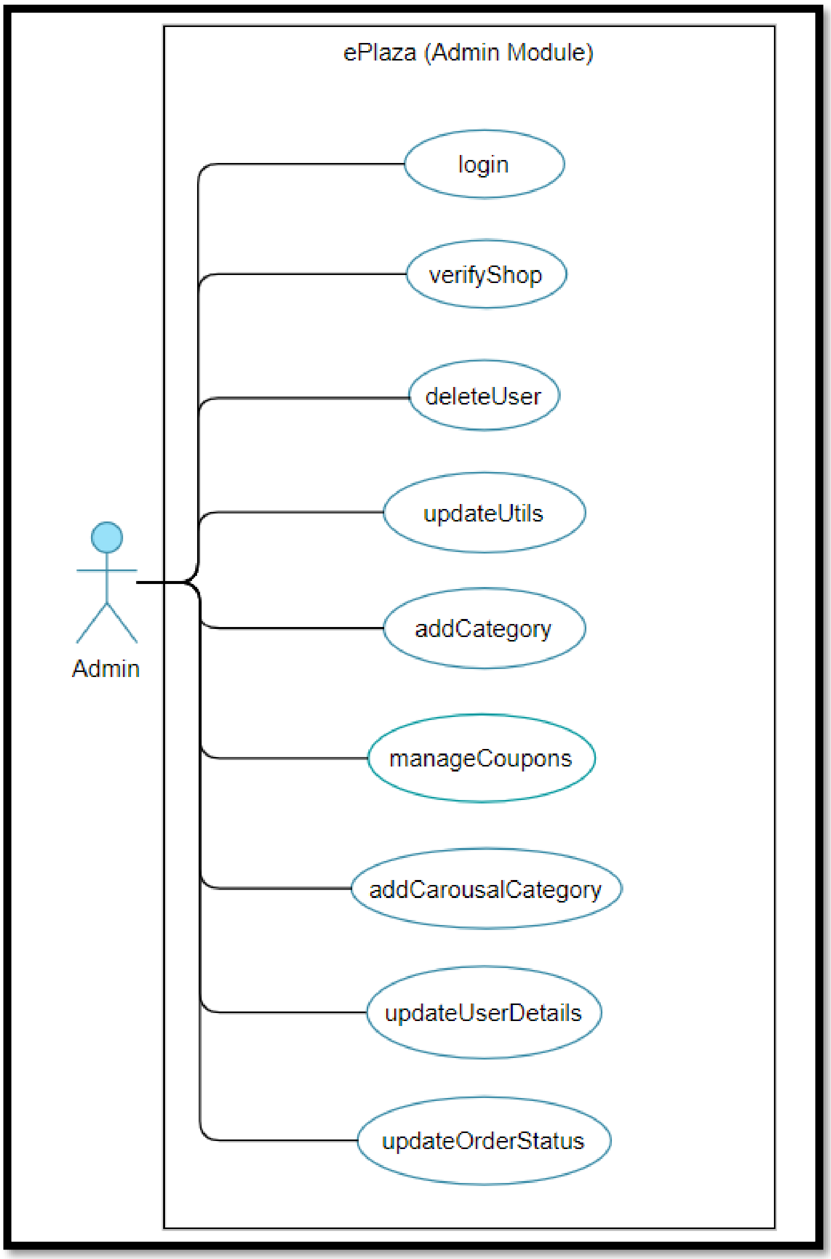


Figure 3:5 Admin Use Case Diagram

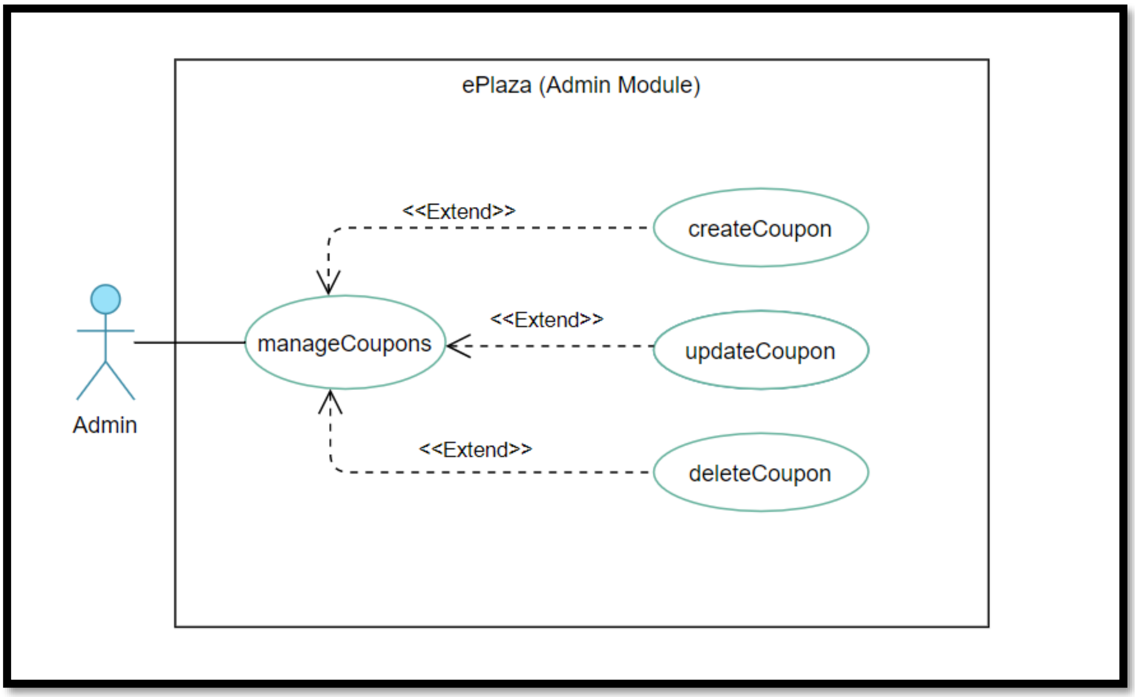


Figure 3:6 Admin Use Case Diagram Manage Coupon

3.2. Functional Requirements

Seller's Functional Requirements

3.2.1. Login for Seller

Table 3:1 Seller Login

Use Case Name			**Login**
Use Case description			Seller login to his profile
Actors			Seller
Pre-condition			Internet, Browser, Username, Password.
Post Condition			Successfully logged in
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter Username, Password		1a) Empty field error
2	Click Login	Logged in	2a) Invalid Credentials

3.2.2. Register for Seller

Table 3:2 Seller Register

Use Case Name			**Register**
Use Case description			Seller can register his account.
Actors			Seller
Pre-condition			Internet, Browser, Email, Phone number.
Post Condition			Mail verification Popup
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter credentials.		1a) Empty field error 1b) Password not matching. 1c) Weak password.
2	Click verify me	Account created	2a) Account not created

3.2.3. Verify Email

Table 3:3 Verify Email

Use Case Name			**Verify Email**
Use Case description			Seller verifies his email to create account by placing correct code in field box.
Actors			Seller
Pre-condition			Internet, Browser, Email address.
Post Condition			Mail verification completed and account created.
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Click Send Code	Clicked	
2	Enter Code	Entered	
3	Click Verify	Verified	3a) not verified

3.2.4. Update Profile

Table 3:4 Seller Update Profile

Use Case Name			**Update profile**
Use Case description			Seller can modify his details
Actors			Seller
Pre-condition			Internet, Browser, Seller logged in
Post Condition			Successfully updated
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter first name, last name, phone.		1a) Empty field error
2	Click update	Updated	2a) Not updated

3.2.5. Create Shop

Table 3:5 Create Shop

Use Case Name			**Create Shop**
Use Case description			Seller can create his shop by entering details.
Actors			Seller
Pre-condition			Internet, Browser, Seller logged in.
Post Condition			Successfully created Shop.
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter Shop name, image, address, Logo, Select Category, phone, Description.		1a) Empty field error
2	Click Create	Created	2a) shop not created.

3.2.6. Update Shop

Table 3:6 Update Shop

Use Case Name			**Update Shop**
Use Case description			Seller can update his shop details
Actors			Seller
Pre-condition			Internet, Browser, Seller profile, Shop
Post Condition			Shop Details Updated
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter new Shop name, Image, Logo, phone, Short Description.		1a) Empty field
2	Click update	Updated	2a) Not updated

3.2.7. Create Product

Table 3:7 Create Product

Use Case Name		**Create Product**	
Use Case description		Seller can add new product to his shop	
Actors		Seller	
Pre-condition		Internet, Browser, Seller Profile, store	
Post Condition		Successfully Added	
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter product name, price, quantity, description, select category or Subcategory, thumbnail, image.		1a) Empty filed.
2	Click Add	Product created	2a) Not created

3.2.8. Update Product

Table 3:8 Update Product

Use Case Name			Update Product
Use Case description			Seller can update product
Actors			Seller
Pre-condition			Internet, Browser, Seller Profile, Shop, Product
Post Condition			Successfully Updated
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter new name, price, Quantity in stock, description, image, and thumbnail.		1a) Empty filed.
2	Click update	Product Updated	2a) Invalid Credentials Error

3.2.9. Delete Product

Table 3:9 Delete Product

Use Case Name			Delete Product
Use Case description			Seller can delete his product
Actors			Seller
Pre-condition			Internet, Browser, Seller profile, Shop, Product
Post Condition			Successfully Deleted
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Select Product	Selected	
2	Click delete	Deleted	2a) Cannot delete

3.2.10. Create Post

Table 3:10 Create Post

Use Case Name			Create Post
Use Case description			A new Post is added to social media page on new product addition.
Actors			Seller
Pre-condition			Internet, Browser, Shop Registered, Social media account.
Post Condition			Successfully Posted
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario**
1	Click Create Product	Post Created at social site	1a) Account not available or token expired. 1b) Not posted.

3.2.11. Update Post

Table 3:11 Update Post

Use Case Name			Update Post
Use Case description			When product's detail is updated, the product's post is updated too.
Actors			Seller
Pre-condition			Internet, Browser, Seller Profile, shop, social media account, Post id.
Post Condition			Successfully Updated
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter new details		
2	Click Update Product	Post is updated at social site.	2a) Token expired. 2b) Not updated.

3.2.12. Delete Post

Table 3:12 Delete Post

Use Case Name			**Delete Post**
Use Case description			The product's post on social media page is also deleted whenever a product is deleted.
Actors			Seller
Pre-condition			Internet, Browser, Seller Profile, shop, social media account, Post id.
Post Condition			Successfully deleted
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Select Product		
2	Click delete product	Post deleted at social site.	2a) Account not available or token expired 2b) Not updated.

3.2.13. Update order Status

Table 3:13 Update Order Status

Use Case Name			**Update Order Status**
Use Case description			Seller can update status of order
Actors			Seller
Pre-condition			Internet, Browser, Shop, Order
Post Condition			Order Status successfully updated
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Select Order	Selected	
2	Change status	Changed	
3	Click Update	Updated	3a) Status not updated.

3.2.14. Update Discount

Table 3:14 Update Discount

Use Case Name			**Update Discount**
Use Case description			Seller can update product discount by adding discount price
Actors			Seller
Pre-condition			Internet, Browser, Seller logged in, Product
Post Condition			Discount successfully updated
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Select product	selected	
2	Enter amount	valid	
3	Click Update	Updated	3a) Discount not updated.

Customer's Functional Requirements

3.2.15. Login for Customer

Table 3:15 Customer Login

Use Case Name			**Login**
Use Case description			Customer login to his profile
Actors			Customer
Pre-condition			Internet, Browser, Username, Password
Post Condition			Successfully logged in
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter Username, Password		1a) Empty field error
2	Click Login	Logged in	2a) Invalid Credentials

3.2.16. Register for Customer

Table 3:16 Customer Register

Use Case Name			Register
Use Case description			Customer profile is profile created
Actors			Customer
Pre-condition			Internet, Browser, Email, Phone
Post Condition			Successfully Account Created
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter username, password, confirm password, phone, email.		1a) Empty fields 1b) Username not available 1c) Email or Phone in incorrect format
2	Click Create	Account Created	2a) Account not created

3.2.17. Recover Password

Table 3:17 Recover Password

Use Case Name			Recover Password
Use Case description			Customer can restore his password
Actors			Customer
Pre-condition			Internet, Browser, Registered
Post Condition			Successfully recovery mail sent.
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter Username, email		1a) Empty field
2	Click Recover Password	Recovery mail sent.	2a) Invalid Username 2b) Email not sent

3.2.18. Update Profile

Table 3:18 Update Profile

Use Case Name			**Update profile**
Use Case description			Customer can modify his details
Actors			Customer
Pre-condition			Internet, Browser, Customer logged in
Post Condition			Successfully updated
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter new details.		1a) Empty field error
2	Click Update	Updated	2a) Profile not updated

3.2.19. Filter Product

Table 3:19 Filter Product

Use Case Name			**Filter Product**
Use Case description			Customer can filter the list of available products based on price, category, brand, and keyword.
Actors			Customer.
Pre-condition			Internet, Browser, Product Filtered.
Post Condition			Successfully filtered.
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Select filter option	Selected	
2	Click Filter button	Filtered result	2a) No Product available

3.2.20. Search Product

Table 3:20 Search Product

Use Case Name			**Search Product**
Use Case description			Customer can search a product through keyword
Actors			Customer
Pre-condition			Internet, Browser, Keyword
Post Condition			Search Result
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Click Search Button	Search drawer open	1a) Keyword empty button disabled
2	Enter keyword	valid	
3	Click Search	Search result	3a) No Product available

3.2.21. Add To Cart

Table 3:21 Add to Cart

Use Case Name			**Add To Cart**
Use Case description			Customer can add product to cart
Actors			Customer
Pre-condition			Internet, Browser, Product
Post Condition			Successfully added to cart
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Select product	selected	1a) Out of stock error
2	Update quantity	Updated	
3	Click Add to cart	Added	3a) Item not added to cart error

3.2.22. Delete From Cart

Table 3:22 Delete from Cart

Use Case Name			**Delete From Cart**
Use Case description			Customer can remove product from cart.
Actors			Customer
Pre-condition			Internet, Browser, Product in cart, Customer logged in.
Post Condition			Successfully removed from cart.
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Select cart	Cart open	
2	Select product	selected	
3	Click Remove	Removed	3a) Not removed

3.2.23. Checkout

Table 3:23 Checkout

Use Case Name			**Checkout**
Use Case description			Customer can move to check out to finalize his cart and total sum with details shown
Actors			Customer
Pre-condition			Internet, Browser, Product in cart, Customer logged in.
Post Condition			Shipping address screen open.
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario**
1	Click checkout option	Transition to shipping screen	

3.2.24. Add Shipping detail

Table 3:24 Add Shipping Details

Use Case Name			**Add Shipping Detail**
Use Case description			Customer enters his shipping details for first time such as address, city.
Actors			Customer
Pre-condition			Internet, Browser, Product in cart, Customer logged in.
Post Condition			Shipping details added
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter Details	valid	a) Empty field error
2	Click Continue	Details added	Not added

3.2.25. Select Payment Method

Table 3:25 Select Payment Method

Use Case Name			**Select Payment Method**
Use Case description			Customer Selects the payment method.
Actors			Customer
Pre-condition			Internet, Browser, Product in cart, Customer logged in.
Post Condition			Payment Method selected
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Select method	Selected	
2	Click continue	Order detail screen	

3.2.26. Place Order

Table 3:26 Place Order

Use Case Name			**Place Order**
Use Case description			Customer can place an order
Actors			Customer
Pre-condition			Internet, Browser, Logged In, Product(s) in cart
Post Condition			Successfully order placed and mail sent.
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Click Place order	Order placed	1a) Order not placed

3.2.27. Track Order

Table 3:27 Track Order

Use Case Name			**Track Order**
Use Case description			Customer can check the status of his order
Actors			Customer
Pre-condition			Internet, Browser, Order
Post Condition			Order Tracking details
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter Order Id	Entered	
2	Click Search	Details	Invalid id

3.2.28. Review Product

Table 3:28 Review Product

Use Case Name			**Review Product**
Use Case description			Customer can add review the product
Actors			Customer
Pre-condition			Internet, Browser, Logged In, Product purchased
Post Condition			Product is reviewed and ratted
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter Comment or Rate	valid	1a) Field empty error
2	Click Post	Posted	2a) Not posted

3.2.29. Give Feedback

Table 3:29 Give Feedback

Use Case Name			**Give Feedback**
Use Case description			Customer can share his thoughts on the feedback form
Actors			Customer
Pre-condition			Internet, Browser, Email Address
Post Condition			Message sent on email
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter credentials	Valid	
2	Enter message	Entered	
3	Click send	Message sent	3a) Mail not sent

3.2.30. Add Comment

Table 3:30 Add Comment

Use Case Name			**Add Comment**
Use Case description			Customer can add comment to his order an optional functionality.
Actors			Customer
Pre-condition			Internet, Browser, Logged In, Order.
Post Condition			Comment added.
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter Comment	Comment added	

3.2.31. Apply Coupon

Table 3:31 Apply Coupon

Use Case Name			**Apply Coupon**
Use Case description			Customer can apply coupon to get discount.
Actors			Customer
Pre-condition			Internet, Browser, Logged In, Order
Post Condition			Coupon is applied
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter Coupon code	Entered	
2	Click Apply	Applied	2a) Invalid coupon

Admin’s Functional Requirements

3.2.32. Login for Admin

Table 3:32 Admin Login

<table>
<tr><th colspan="3">Use Case Name</th><th>Login</th></tr>
<tr><td colspan="3"><b>Use Case description</b></td><td>Admin logins to his profile</td></tr>
<tr><td colspan="3"><b>Actors</b></td><td>Admin</td></tr>
<tr><td colspan="3"><b>Pre-condition</b></td><td>Internet, Browser, Username, Password</td></tr>
<tr><td colspan="3"><b>Post Condition</b></td><td>Successfully logged in</td></tr>
<tr><th>No</th><th>Main Scenario</th><th>Main Scenario Response</th><th>Alternative Scenario Response</th></tr>
<tr><td>1</td><td>Enter Username, Password</td><td></td><td>1a) Empty field error</td></tr>
<tr><td>2</td><td>Click Login</td><td>Logged in</td><td>2a) invalid credentials</td></tr>
</table>

3.2.33. Verify Shop

Table 3:33 Verify Shop

<table>
<tr><th colspan="3">Use Case Name</th><th>Verify Shop</th></tr>
<tr><td colspan="3"><b>Use Case description</b></td><td>Admin has authority to verify the registered shop, and, on his approval, shop is accessible to customer as well as seller.</td></tr>
<tr><td colspan="3"><b>Actors</b></td><td>Admin</td></tr>
<tr><td colspan="3"><b>Pre-condition</b></td><td>Internet, Browser, Admin Account, Shop</td></tr>
<tr><td colspan="3"><b>Post Condition</b></td><td>Shop Status Updated.</td></tr>
<tr><th>No</th><th>Main Scenario</th><th>Main Scenario Response</th><th>Alternative Scenario Response</th></tr>
<tr><td>1</td><td>Select shop</td><td>selected</td><td></td></tr>
<tr><td>2</td><td>Update verification status</td><td>Shop status updated</td><td>2a) Not updated</td></tr>
</table>

3.2.34. Delete User

Table 3:34 Delete User

Use Case Name			Delete User
Use Case description			Admin has authority to delete user account.
Actors			Admin
Pre-condition			Internet, Browser, Admin Account.
Post Condition			Account deleted
No	**Main Scenario**		**Alternative Scenario**
1	Select User	Selected	
2	Click Delete	Deleted	2a) Not deleted

3.2.35. Add Category

Table 3:35 Add Category

Use Case Name			Add Category
Use Case description			Admin can add a new category
Actors			Admin
Pre-condition			Internet, Browser, Admin account.
Post Condition			Category added
No	**Main Scenario**		**Alternative Scenario**
1	Select level	Selected	
2	Select parent category	Selected	2a) No parent category
3	Enter category name	Entered	
4	Click Add	Added	4a) Not added

3.2.36. Update Utils

Table 3:36 Update Utils

Use Case Name			**Update Utils**
Use Case description			Admin can update the utils.
Actors			Admin
Pre-condition			Internet, Browser, Admin Account.
Post Condition			Account deleted
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter new utils value		Empty fields
2	Click Update	Updated	2a) Not Updated

3.2.37. Update User Details

Table 3:37 Update User Details

Use Case Name			**Update User Details**
Use Case description			Admin can update user details this can be name, password, or email.
Actors			Supper Admin.
Pre-condition			Internet, Browser, Admin account, User.
Post Condition			Successfully updated
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Enter new details		1a) Empty fields error
2	Click update	Updated	2a) Not updated

3.2.38. Update Order Status

Table 3:38 Update Order Status

Use Case Name			Update Order Status
Use Case description			Admin can update order status in terms of canceling the order
Actors			Admin
Pre-condition			Internet, Browser, Admin Account, Order.
Post Condition			Successfully updated
No	**Main Scenario**		**Alternative Scenario**
1	Select order	Selected	
2	Change status	Changed	
3	Click update	Updated	3a) Not updated

3.2.39. Add Carousel Category

Table 3:39 Add Carousel Category

Use Case Name			Add Carousel Category
Use Case description			Admin can add a category for slider at home screen to show top selling category products
Actors			Admin
Pre-condition			Internet, Browser, Admin account, Category.
Post Condition			Successfully Added
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Select category	Selected	
2	Click add	Added	Not Added

3.2.40. Create Coupon

Table 3:39 Create Coupon

<table>
<tr><td colspan="3"><b>Use Case Name</b></td><td><b>Create Coupon</b></td></tr>
<tr><td colspan="3"><b>Use Case description</b></td><td>Admin can add a new coupon.</td></tr>
<tr><td colspan="3"><b>Actors</b></td><td>Admin</td></tr>
<tr><td colspan="3"><b>Pre-condition</b></td><td>Internet, Browser, Shop</td></tr>
<tr><td colspan="3"><b>Post Condition</b></td><td>Successfully Added</td></tr>
<tr><td><b>No</b></td><td><b>Main Scenario</b></td><td><b>Main Scenario Response</b></td><td><b>Alternative Scenario Response</b></td></tr>
<tr><td>1</td><td>Enter coupon code and percentage</td><td></td><td>Empty Field Error</td></tr>
<tr><td>2</td><td>Click Create</td><td>Created</td><td>Coupon code already in use error.</td></tr>
</table>

3.2.41. Update Coupon

Table 3:40 Update Coupon

<table>
<tr><td colspan="3"><b>Use Case Name</b></td><td><b>Update Coupon</b></td></tr>
<tr><td colspan="3"><b>Use Case description</b></td><td>Admin can update coupon details</td></tr>
<tr><td colspan="3"><b>Actors</b></td><td>Admin</td></tr>
<tr><td colspan="3"><b>Pre-condition</b></td><td>Internet, Browser, Shop, Coupon.</td></tr>
<tr><td colspan="3"><b>Post Condition</b></td><td>Successfully updated</td></tr>
<tr><td><b>No</b></td><td><b>Main Scenario</b></td><td><b>Main Scenario Response</b></td><td><b>Alternative Scenario Response</b></td></tr>
<tr><td>1</td><td>Enter new details.</td><td></td><td>1a) Empty Fields Error</td></tr>
<tr><td>2</td><td>Click Update</td><td>Updated</td><td>2a) Not updated</td></tr>
</table>

3.2.42. Delete Coupon

Table 3:41 Delete Coupon

Use Case Name			**Delete Coupon**
Use Case description			Admin can delete coupon.
Actors			Admin
Pre-condition			Internet, Browser, Shop, Coupon.
Post Condition			Successfully Deleted
No	**Main Scenario**	**Main Scenario Response**	**Alternative Scenario Response**
1	Select Coupon	selected	
2	Click delete	Deleted	2a) Not deleted

3.3. Non-Functional Requirements

3.3.1. Security Requirements

Keeping system secure and information authentic tends to be high priority in our site, to build trust with Seller and Customer. Following is level of security taken to keep database secure.

- The database information such as name, Password, Port in settings.py is set through environment variables to keep it hidden from unwanted user.
- At hosting server-side SSL certificates for secure web browsing at ePlaza is provided.
- Apart from this Login and Registration is secured by Django tokenization (JWT).

3.3.2. Adaptability

Bootstrap and Material UI is applied on front to keep the site responsive, so site is adaptable on any screen. Along media quarries to make screens responsive on different mobile screen.

3.3.3. Availability

System is being deployed on Haruko server, so it is available all the time, also exception handling is done on all pages to keep system available to use even if exception occurs.

3.3.4. Flexibility

Django allows us to construct as versatile a system as feasible. ePlaza includes the User, Orders, Products, and Shop apps, each of which operates independently. The rest framework separates frontend and backend so that we may simply add and remove new frontend frameworks in the future. Not only that, from frontend side, screens are built from components, making UI features replacement simple.

3.3.5. Interoperability

Through rest framework the advantage is that we can maintain same database and backend business logic for different devices such as android and web, the communication is done through API calls.

3.3.6. Maintainability

The System is easy to maintain due to independency of Screens, Components, and views, any new screen in frontend with its backend logic is implementable without impacting other modules. Along with impact of change is kept minimum as possible by keeping each component separate.

3.3.7. Portability

The site is responsive so can be accessed on mobile devices to attain portability.

3.3.8. Reliability

On every order placement status is updated by seller and customer is notified with at its profile, cart is updated on every time on push of add to cart button and so on, screen and components consistency is maintained thus system tends to be reliable.

3.3.9. Usability

Keeping in mind ePlaza is for both naïve user such as local shopkeeper, housewife's, and children's etc. So, system interface and interaction are kept simple and consistence as possible.

3.3.10. Technology Constraint

From technology point of view, we restricted to following requirements

- Backend Django.
- Front React JS
- Database Postgres
- AWS buckets
- Hosting Server Heroku

3.4. Interface Requirements

3.4.1. UI Requirements

Following are frontend UI frameworks used in our system.

- MUI (Material User Interface)
- React Bootstrap

3.4.2. Hardware Interface

No such hardware component is used in our system.

3.4.3. Software Interface:

- Frontend developed in React.
- Backend developed in Django Rest Framework for business logic.
- Postgres database used as relational database.
- AWS buckets to store images.

3.4.4. Communication Interface

- HTTP
- TCP/IP
- UDP

3.5. Resource Requirement

3.5.1. Hardware Requirements

Table 3:42 Hardware Specs

Recommended RAM	**512 MB**
Min Disk Space	150 MB (XP), 70 (Vista)
Rec. Disk Space	70 Mb
Rec. Disk Space 64-bit	120 MB
Browser	Chrome, Edge

3.5.2. Hardware Devices

As site is responsive so can viewed on any device such as

- Laptop
- Mobile phone
- Tablets

3.6. Database Requirements

- Heroku Postgres database is used to map Database models using Django ORM.
- Amazon bucket is used for storing images.
- For database connectivity Postgres version 12 is used.
- Other than that local storage and Session storage is required to store user details.

3.7. Project Feasibility

3.7.1. Technical Feasibility

This System is a Web Application. We have developed the system in such a way that it can Cope with the difficulty every user face while ordering something online. Every other website is using complex interface which makes it difficult for everyday user to get to the final step without any hurdle. security and risk threads. We have designed the system Using Model View Template (MVT) architecture to be scalable at any time. We have developed the system under current conditions, requirements in the digital sector of Pakistan, which mainly targets the e-commerce of Pakistan, and our System is highly recommended for practical use in the corresponding industry of our country. Keeping in mind that the bigger giants in this industry are acquired by foreign investors and Pakistan do not has any big fish which is made in Pakistan and made for Pakistan.

3.7.2. Operational Feasibility

We have developed the system in such a way that it satisfies all the requirements that makes it user friendly. The system can provide all the services which are included in an e-commerce platform. Our platform allows the user to get their hands on a product directly by clicking on the "Add to cart" icon. Moreover, the system offers adequate controls to the user to get discounts and deals over number of items through coupons.

In addition to this, the system is highly scalable. It is because we've acquired the expertise of M&P courier service which delivers to almost every part of Pakistan and help us to scale at a larger level.

3.7.3. Legal and Ethical Feasibility

Our system fulfils its legal and ethical aspects. It cannot be used without proper authorization. The system provides security to the data, and without authorization, the data is not displayed. The system doesn't involve any device that can be object able. Moreover, the system is easily adaptable and can be configured easily. For any user, if they use this system, it will provide more extensive data security. The system defines the System usage, terms and conditions, and privacy policies for their respective user. However, the system tends to have its privacy policy.

3.8. Conclusion

At the end of the chapter the system functional and non-functional requirements are clear, however other requirements related hardware, software and database are also known. Analysis model and feasibility study add further value to understanding system business purpose. Overall, the system scope in known by the end.

Chapter 4

SYSTEM DESIGN

4.1. Design Approach

4.1.1. About design

The System design is based on Django server at the back end and React at the front-end following diagram shows the flow between them

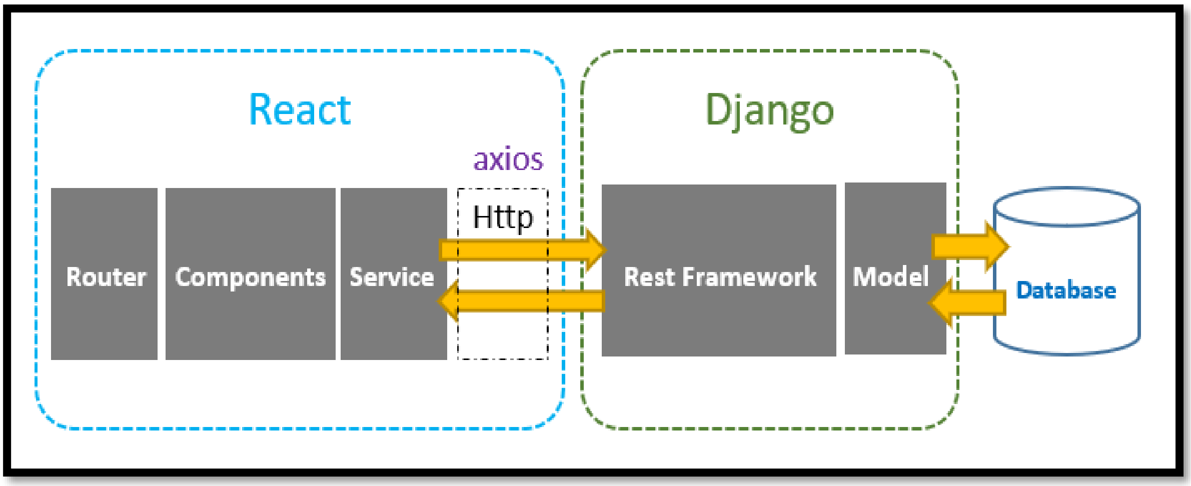


Figure 4:1 Django React Flow Diagram

4.1.2. React JS

React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets us to compose complex UIs from small and isolated pieces of code called “components” this makes our UI implementation fast without redundancy. However, our system involves function-based methods with react

- Hooks
- States
- Use effects
- History etc.

Making development more effective and efficient. For Asynchronous Calls from front to back axios are used.

4.1.3. Rest Framework

The Django REST framework is a versatile and robust toolkit for creating Web APIs. We were able to work with APIs by translating data in JSON and incorporating CRUD functionality via GET, POST, DELETE, and PUT commands in the rest framework. As seen below

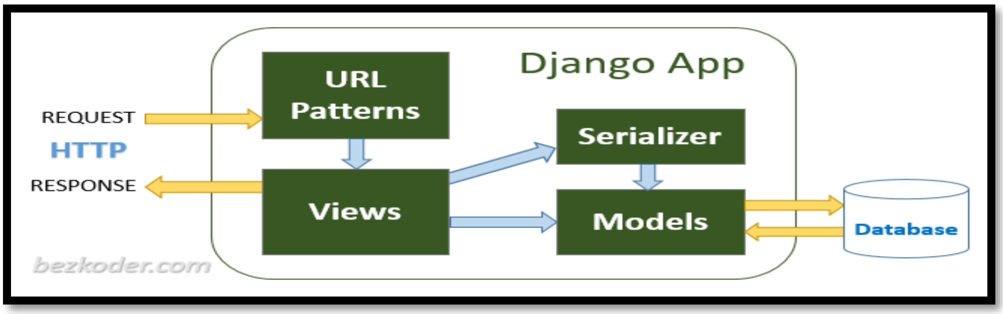
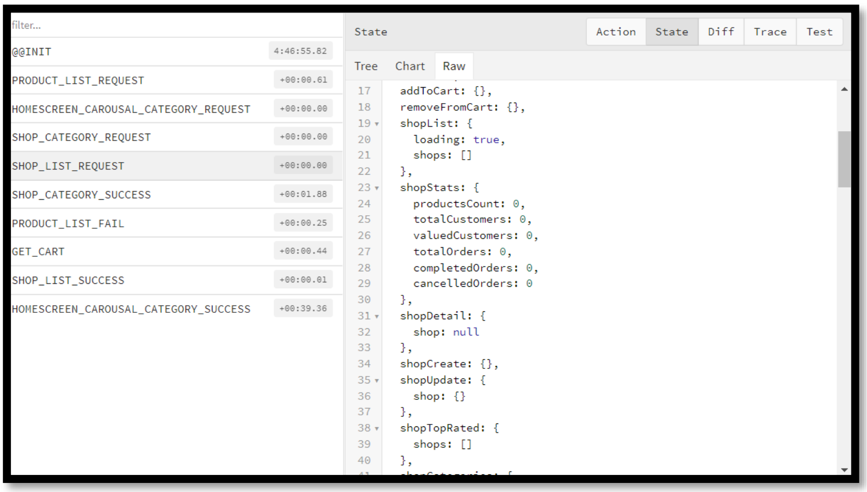


Figure 4:2 Django App Flow Diagram

4.1.4. Redux

Redux, an open-source library for managing and centralising application states, is used at the front to handle communication between screens and components in terms of error, data, and loading. Our data at the front, on the other hand, is saved in redux states from whence we call it when we need it. It also facilitates debugging. The redux states are depicted in the diagram below.



```
@@INIT                                   4:46:55.82
PRODUCT_LIST_REQUEST                     +00:00.61
HOMESCREEN_CAROUSAL_CATEGORY_REQUEST     +00:00.00
SHOP_CATEGORY_REQUEST                    +00:00.00
SHOP_LIST_REQUEST                        +00:00.00
SHOP_CATEGORY_SUCCESS                    +00:01.88
PRODUCT_LIST_FAIL                        +00:00.25
GET_CART                                 +00:00.44
SHOP_LIST_SUCCESS                        +00:00.01
HOMESCREEN_CAROUSAL_CATEGORY_SUCCESS     +00:39.36
```

```
  addToCart: {},
  removeFromCart: {},
  shopList: {
    loading: true,
    shops: []
  },
  shopStats: {
    productsCount: 0,
    totalCustomers: 0,
    valuedCustomers: 0,
    totalOrders: 0,
    completedOrders: 0,
    cancelledOrders: 0
  },
  shopDetail: {
    shop: null
  },
  shopCreate: {},
  shopUpdate: {
    shop: {}
  },
  shopTopRated: {
    shops: []
  },
```

Figure 4:3 Redux States

4.1.5. About File Structure

ePlaza file structure is a planned structure this is classified as

4.1.5.1. Backend Block Diagram

Following diagram shows the backend structure.

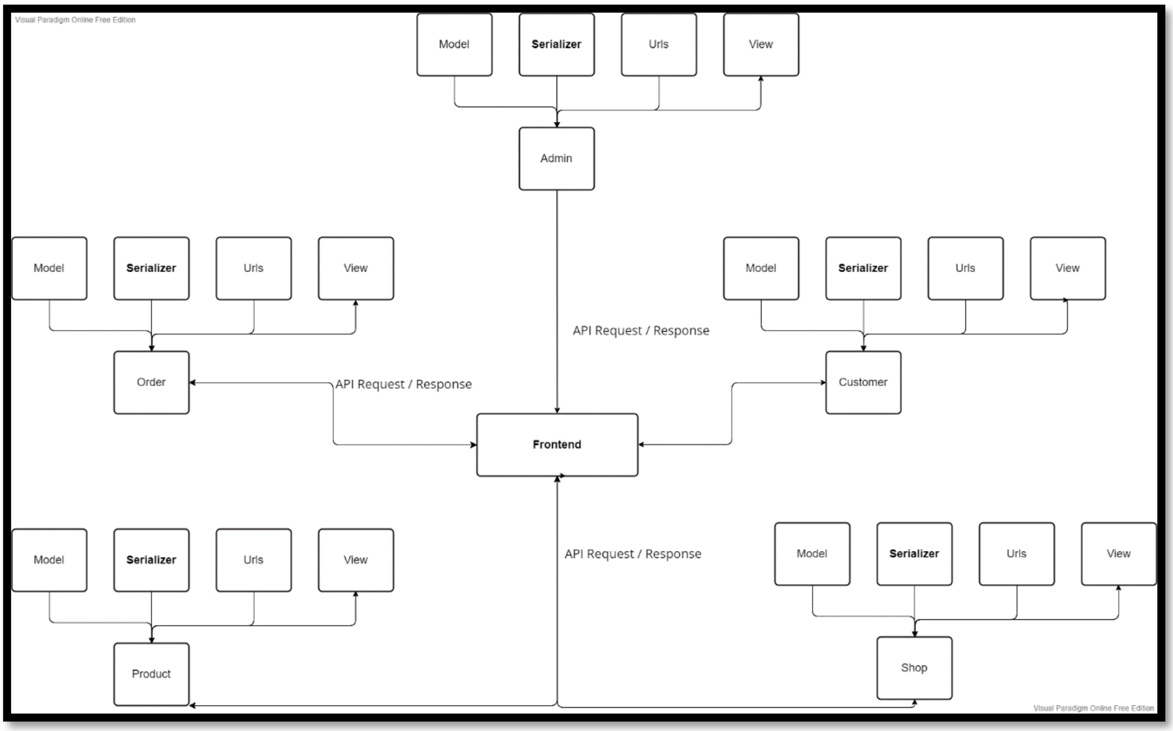


Figure 4:4 Backend Block Diagram

Table 4:1 Backend Block diagram description

NAME	**DESCRIPTION**
Order	Handling all logic for functions related to customer orders and status management to Tracking.
Product	Handling all logic for functions related to Product create, update, delete and review etc.
Shop	Handling all logic for functions related to Shop create update and shop clicks events etc.

Customer	Handling all logic for functions related to Customer profile create, update, delete and level management etc.
Admin	Handling all logic for functions related to Super Admin such as shop verification, category management.
View	File that holds the business logic
Serializers	File that converts query sets or model object into JSON format.
URL’s	File that holds URL paths and map them to function present in View file. Frontend calls are directed from URLs to View
Model	File that holds all database models.

4.1.5.2. Frontend Block Diagram

Following diagram shows the frontend structure.

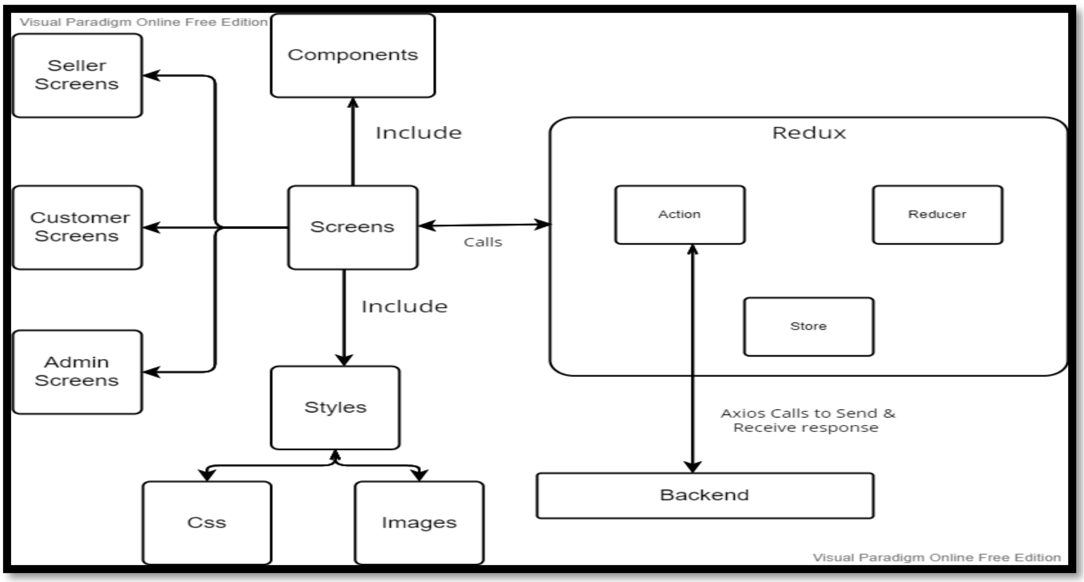


Figure 4:5 Frontend Block Diagram

Table 4:2 Frontend Block diagram description

NAME	DESCRIPTION
Screens	Screens holds the design elements of website including Html, inline CSS, use states, Use effects etc.
Components	Includes elements that supports screen to design, component’s main purpose to satisfy usability discard duplication such Cards, Header and Sidebar etc. are all components
Shop	Style file is used to capture the images and CSS of screens or components.
Action	The store's only source of information is actions. Actions have a field that specifies the sort of action to be performed and sends requests to Back in the form of POST, PUT, DELTE, and GET.
Reducer	A reducer is a pure function that accepts an action and the application's previous state and returns the new state. The action defines what occurred, and it is the reducer's responsibility to return the new state as a result of that action.
Store	A store is a state container that holds the state of the application. In our application, Redux has one store where all states are saved.

4.2. System Architecture

The System Architecture is based on MVT (Model View Template).

It is a software design pattern. Three important components are involved which are Model, View, and Template. This architecture does not involve a separate controller and the complete application is based on Model View and Template. The Model helps to handle the database. which is a data access layer that handles the data.

In our case, we'll not be using Template since we've no UI instead frontend framework (React JS), this handles our all-UI functions and components which communicate through API calls. The diagram below shows the flow.

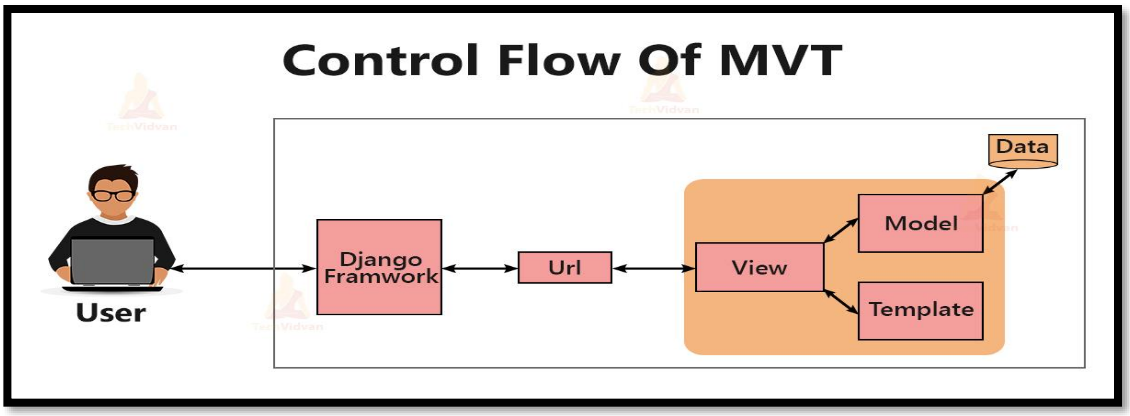


Figure 4:6 MVT Flow Diagram

4.2.1. Model

This is responsible for maintaining data, through ORM the tables are mapped in Database from Models. It is the logical data structure behind the entire application and is represented by a database (generally relational databases such as Postgres).

4.2.2. View

This is responsible for handling business logic, these are called on visiting specific URL and returns json back to frontend on processing, Serializers are involved here to convert object to json vice versa.

4.2.3. Template

All Html files are handled by Templates but in our case Frontend framework is involved to handle our UI functionalities and rendering of templates as shown in figure 4.6.

4.2.4. URL

URLs are responsible for mapping the frontend call to business logic present in View. The URL path is provided to each function declared in URL file on matching of that path corresponding function is called.

4.3. Logical Design

4.3.1. Class Diagram

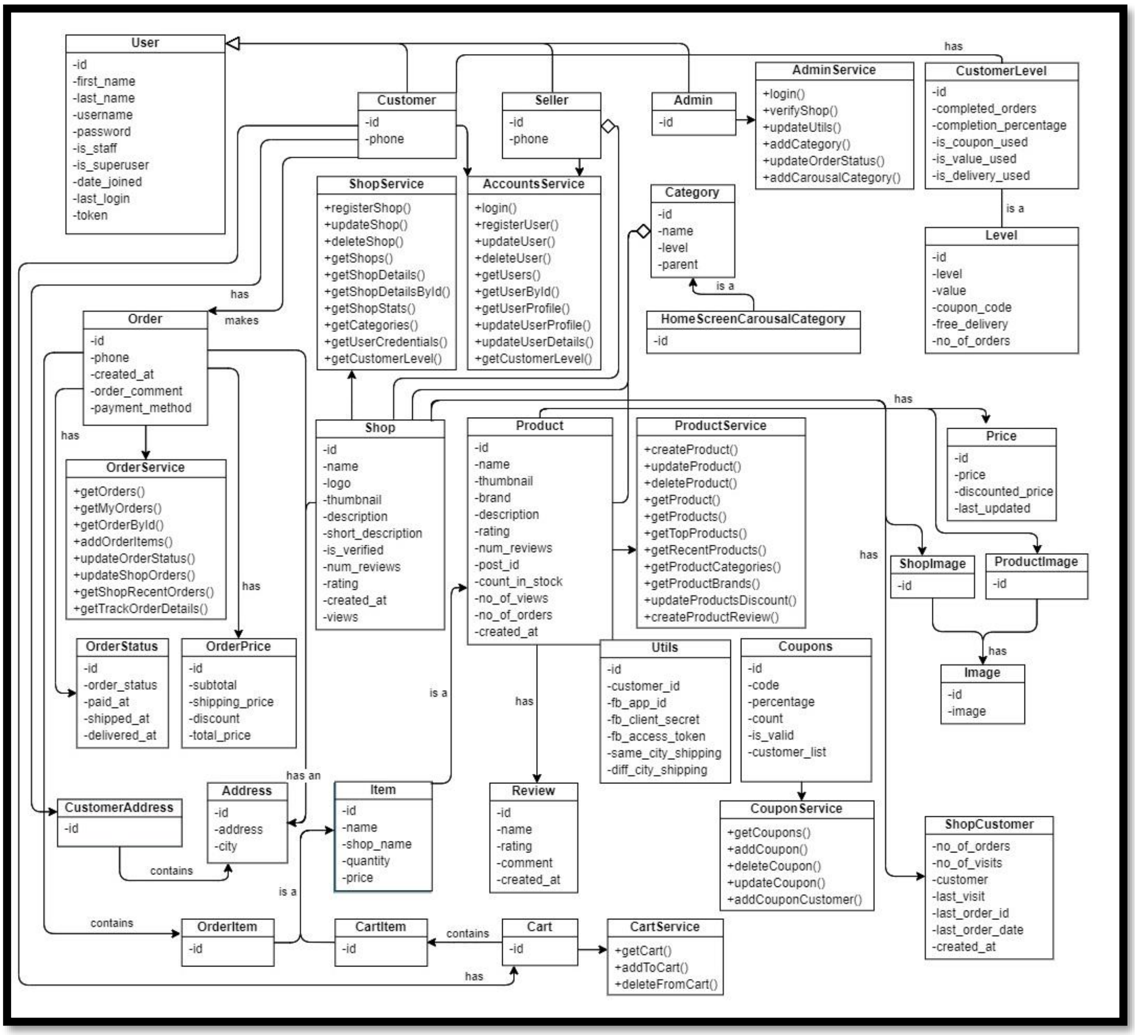


Figure 4:7 Class Diagram

4.4. Dynamic View

4.4.1. Sequence Diagram Format:

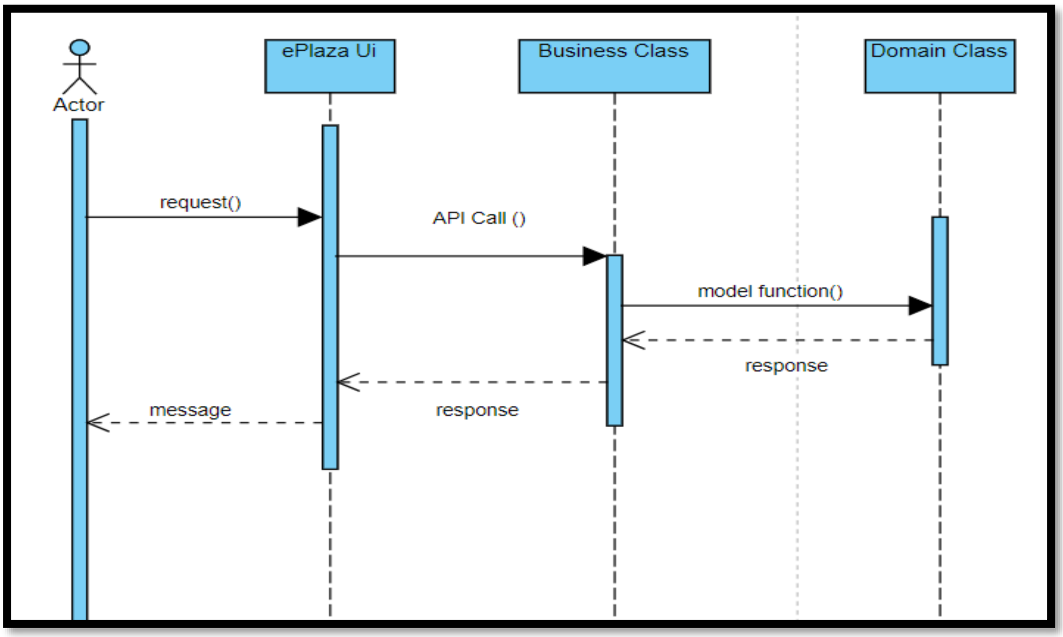


Figure 4:8 Sequence diagram format with MVT

ePlaza UI (Template):

A frontend UI screen for the actor to interact with the functionality of system such as enter text, click button, etc.

Business class (View):

The class that holds the business logic of a system, is invoked through a frontend API call. These can be functions present in the service file of python. Such as login, register and add product etc.

Domain class (Model):

Holds the function related to the database or model such as create, save, delete, put etc. by invoking such function we can alter or get database objects with support of ORM (Object Relational Mapping).

4.4.2. Seller Sequence Diagrams

4.4.2.1. Login for Seller

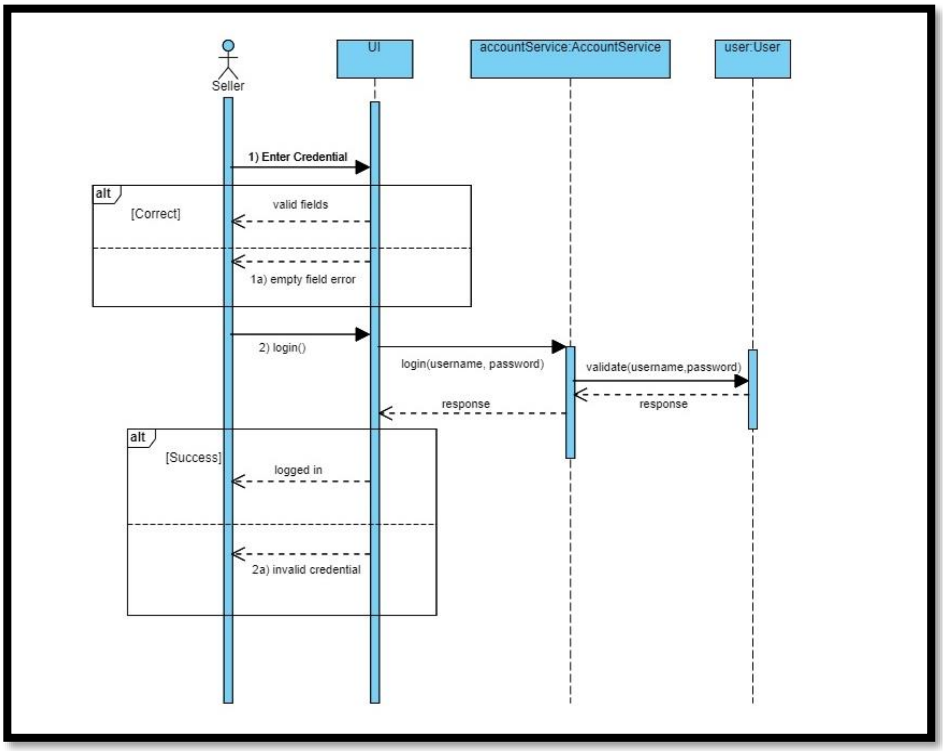


Figure 4:9 Seller Login

4.4.2.2. Register

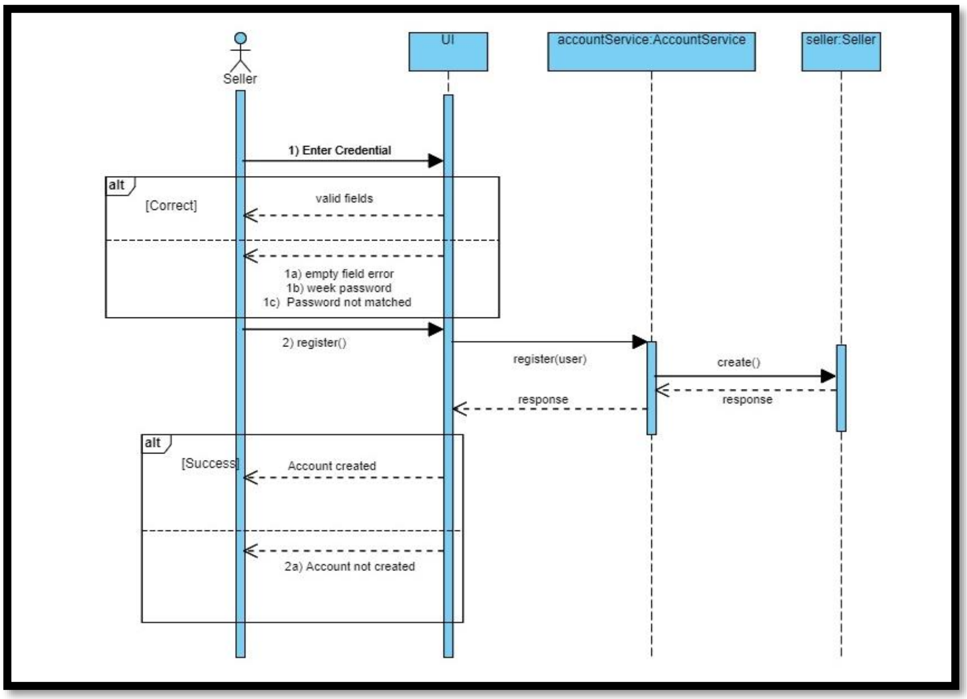


Figure 4:10 Seller Register

4.4.2.3. Verify Email

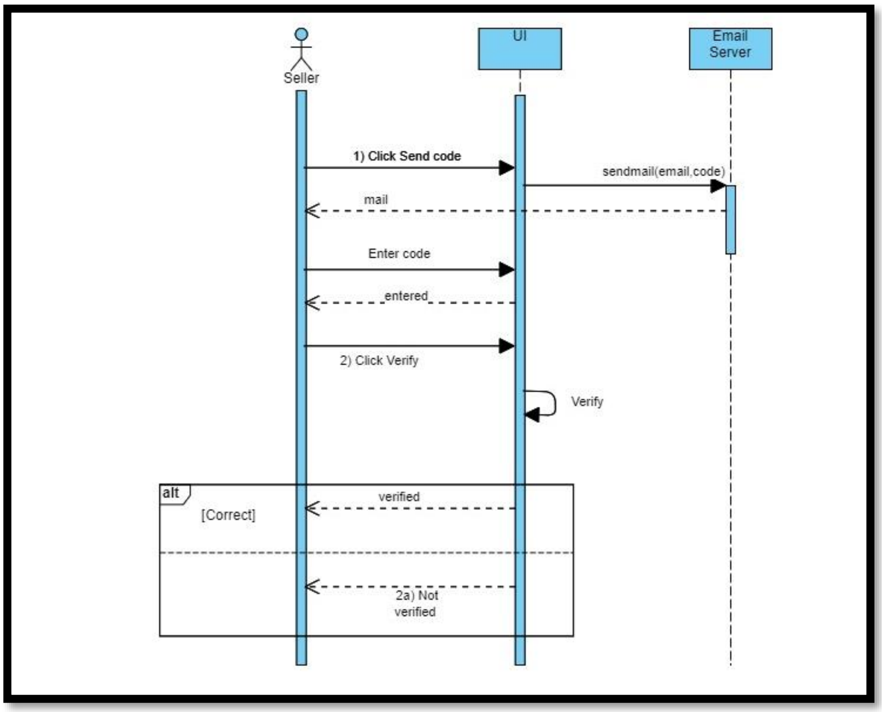


Figure 4:11 Verify Email

4.4.2.4. Update Profile

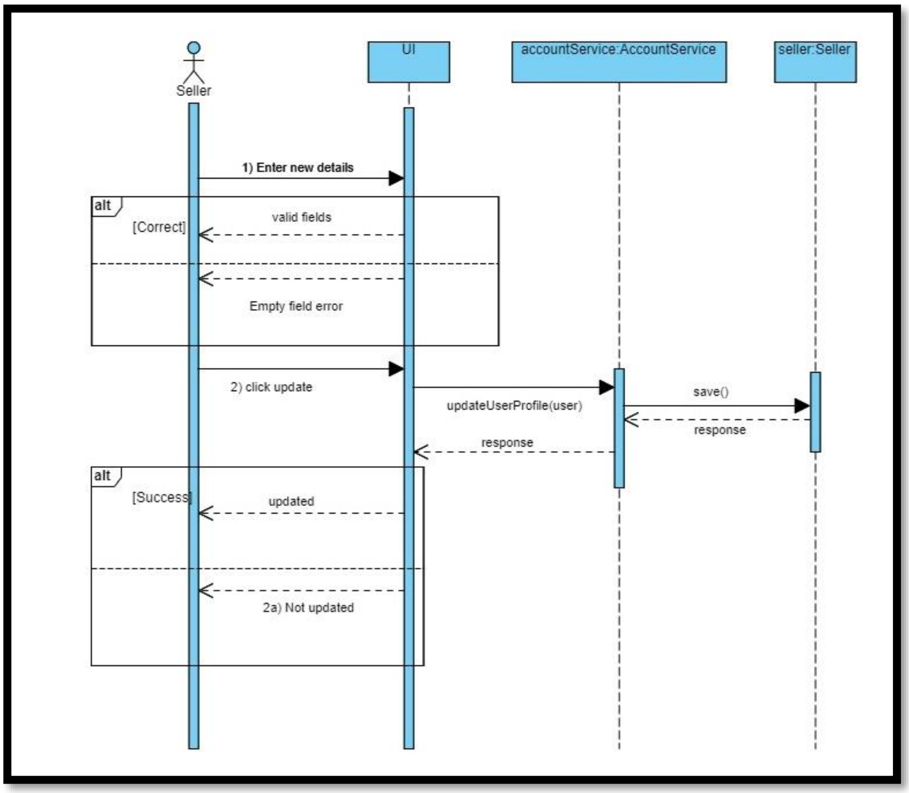


Figure 4:12 Update Profile

4.4.2.5. Create Shop

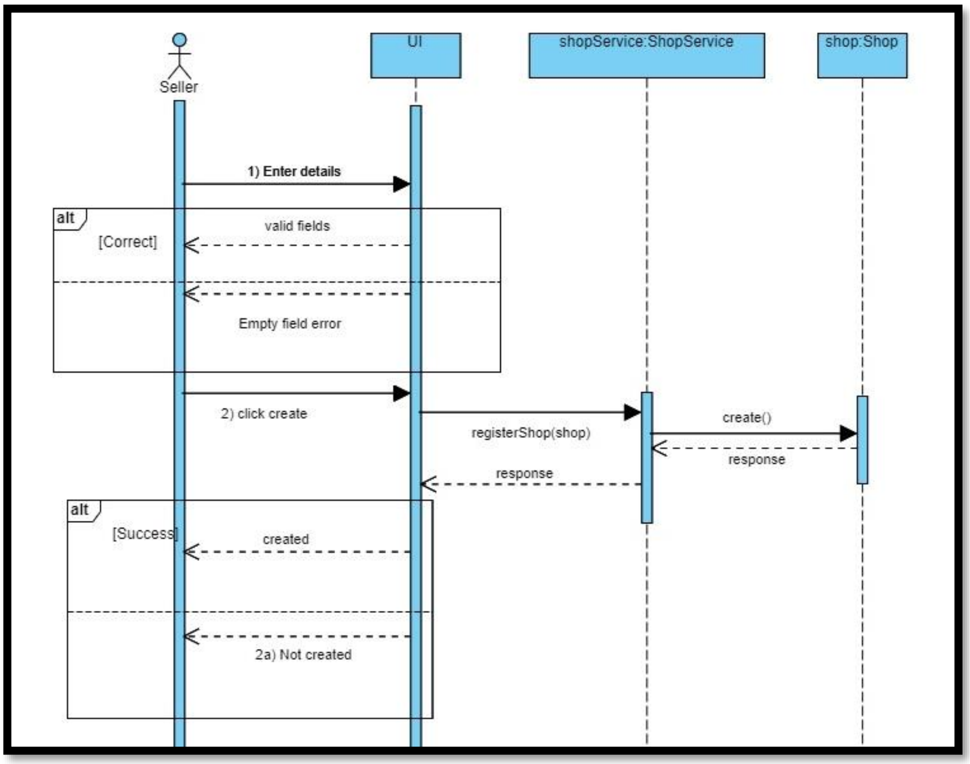


Figure 4:13 Create Shop

4.4.2.6. Update Shop

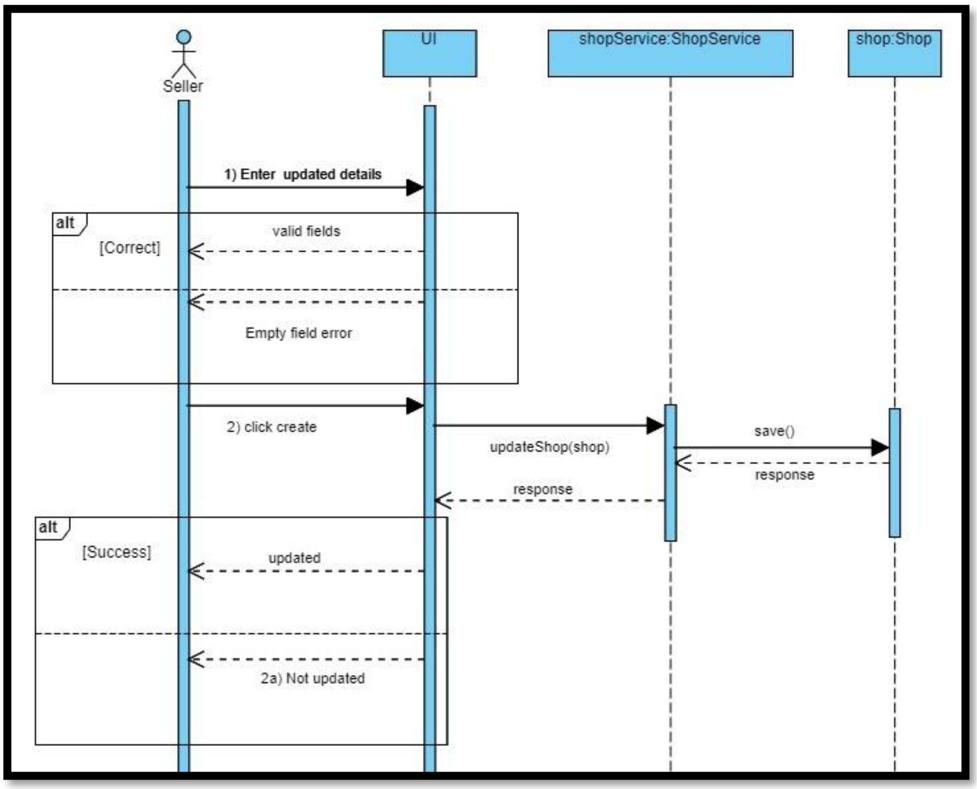


Figure 4:14 Update Shop

4.4.2.7. Create Product

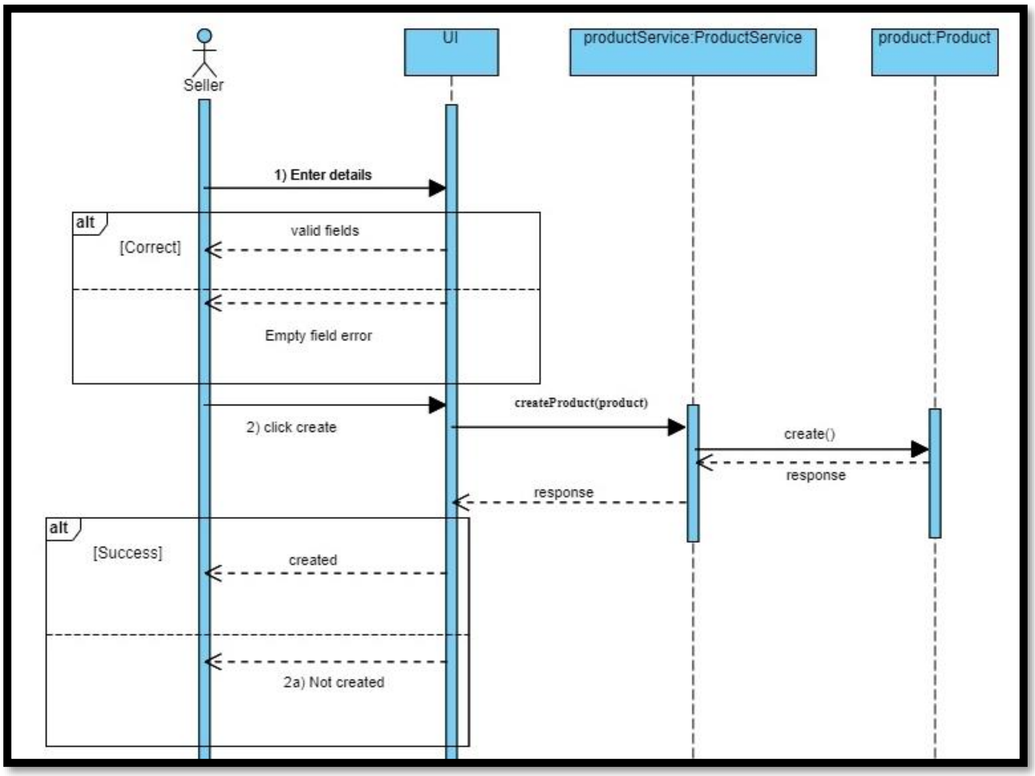


Figure 4:15 Create Product

4.4.2.8. Update Product

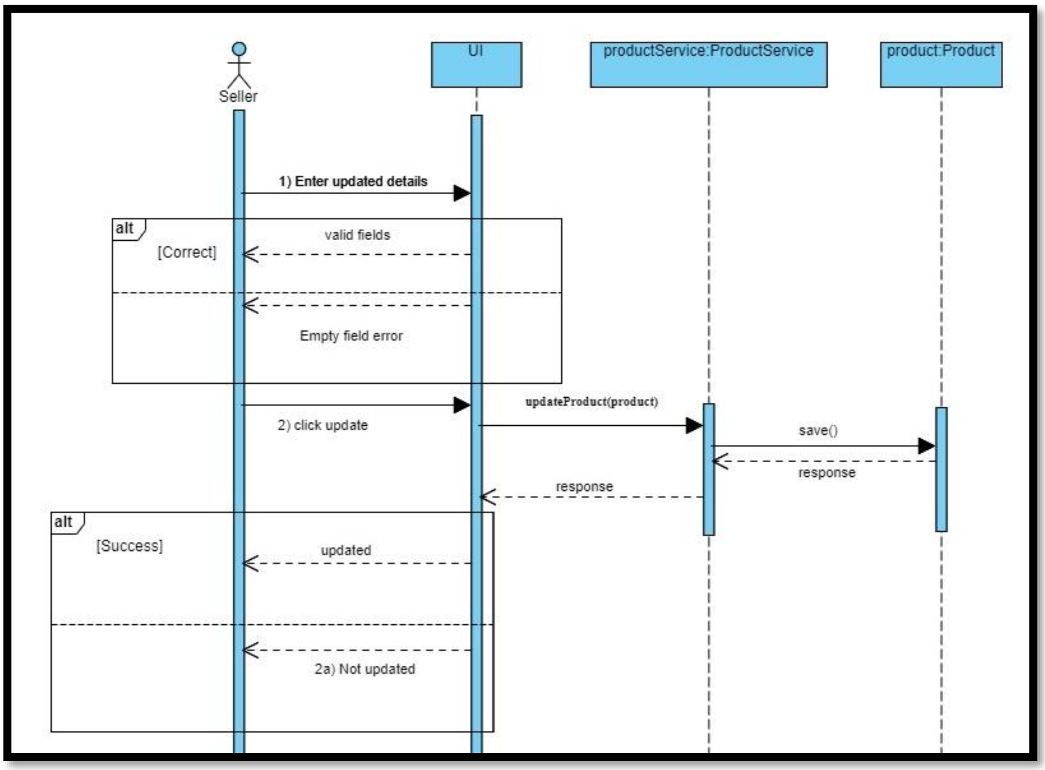


Figure 4:16 Update Product

4.4.2.9. Delete Product

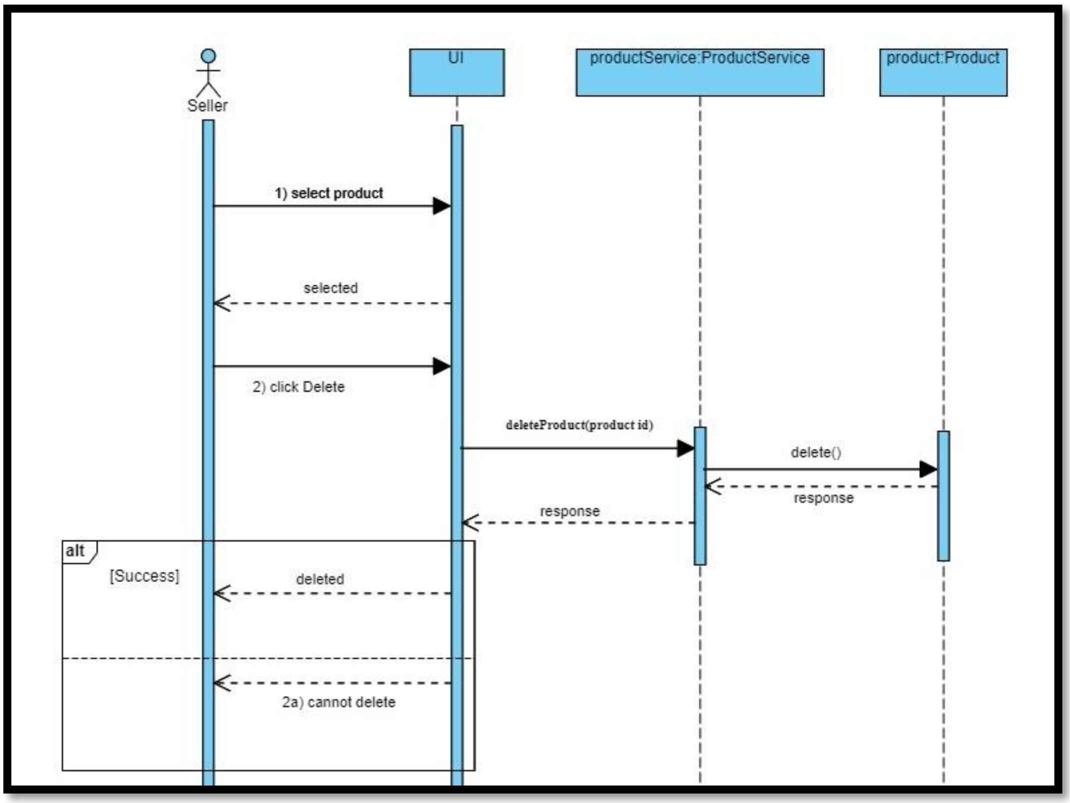


Figure 4:17 Delete Product

4.4.2.10. Create Post

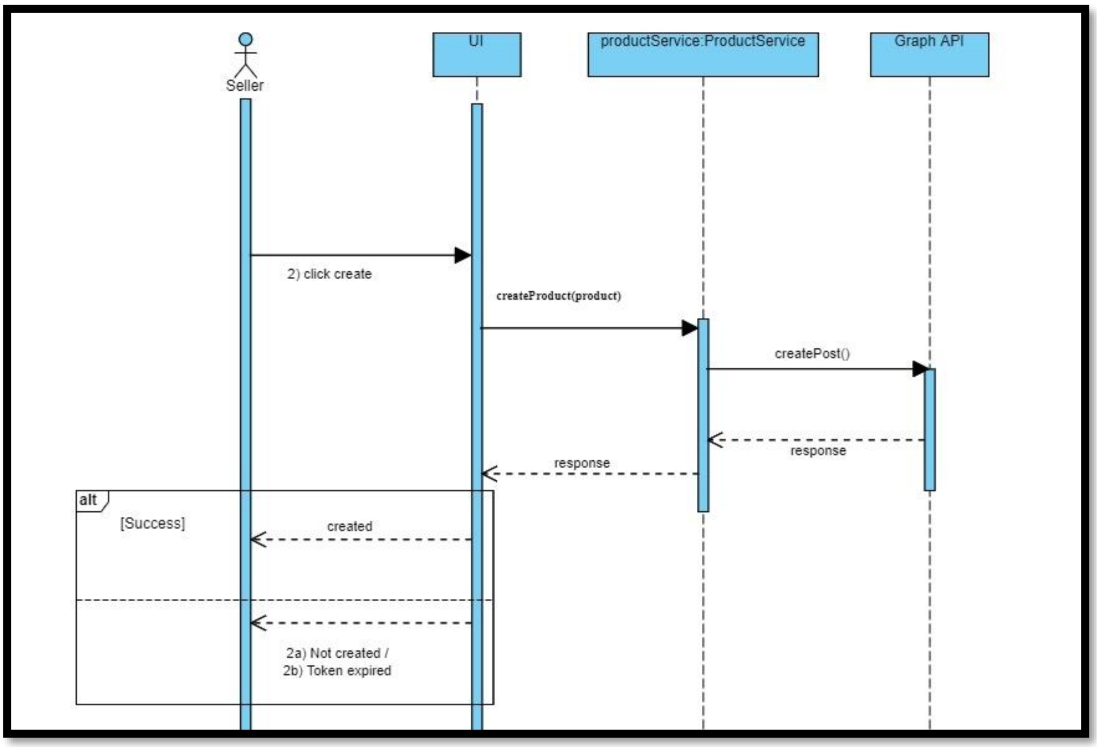


Figure 4:18 Create Post

4.4.2.11. Update Post

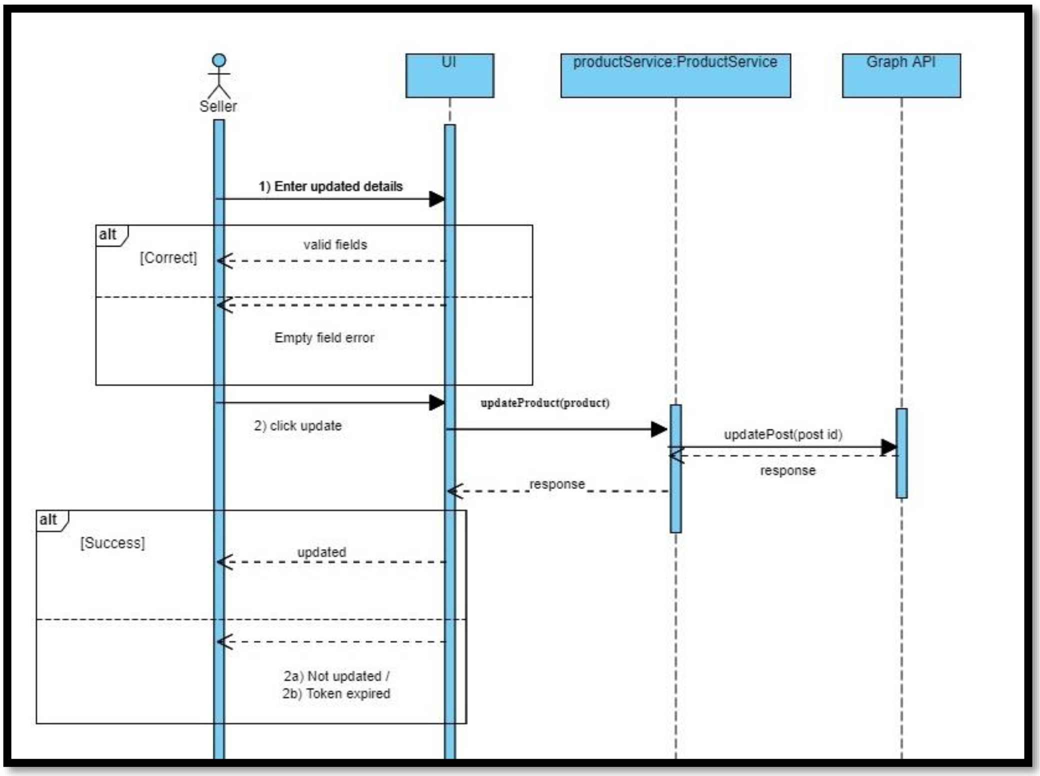


Figure 4:19 Update Post

4.4.2.12. Delete Post

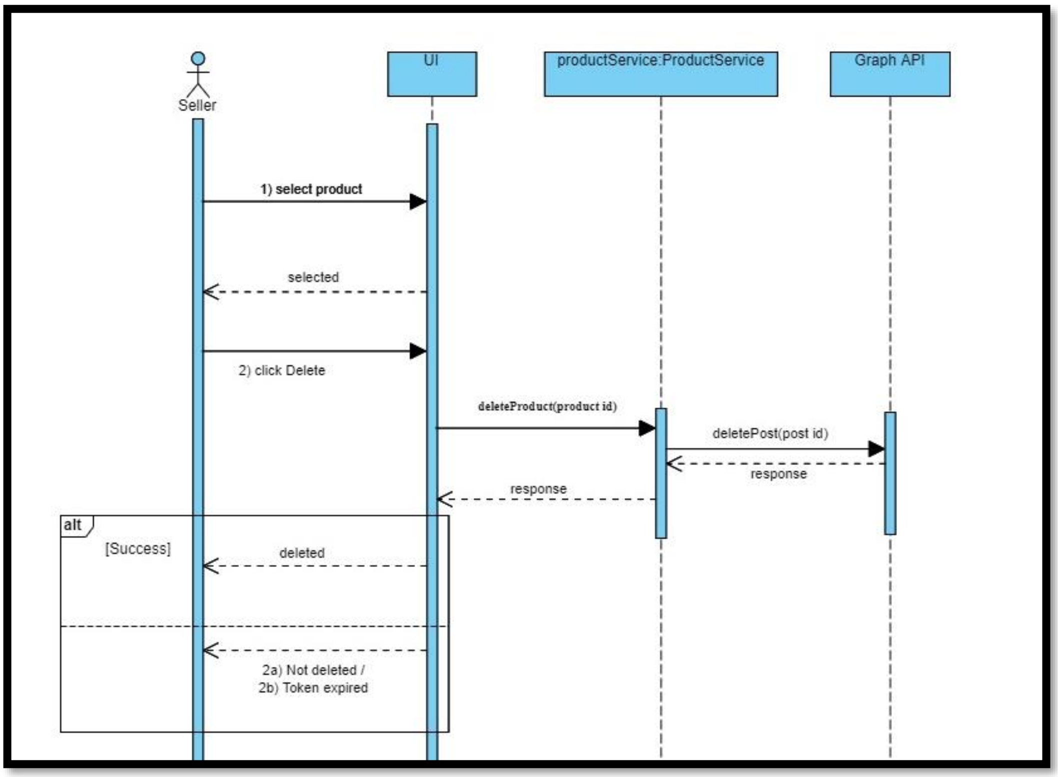


Figure 4:20 Delete Post

4.4.2.13. Update order Status

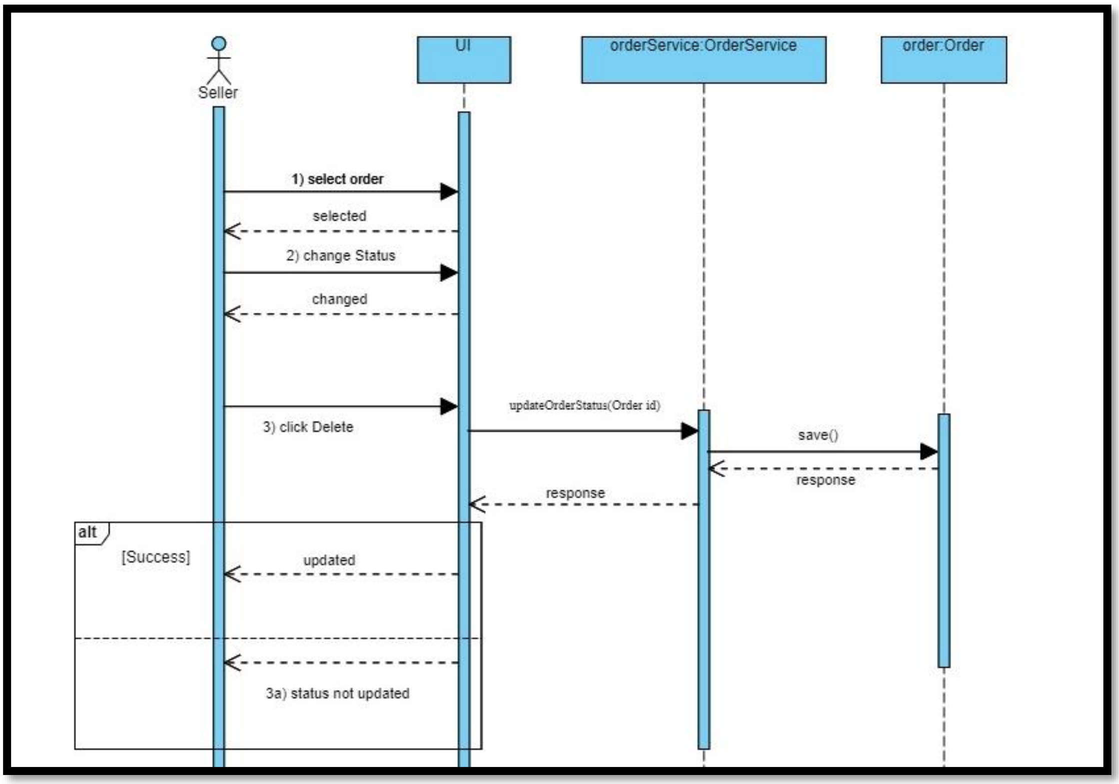


Figure 4:21 Update Order Status

4.4.2.14. Update Product Discount

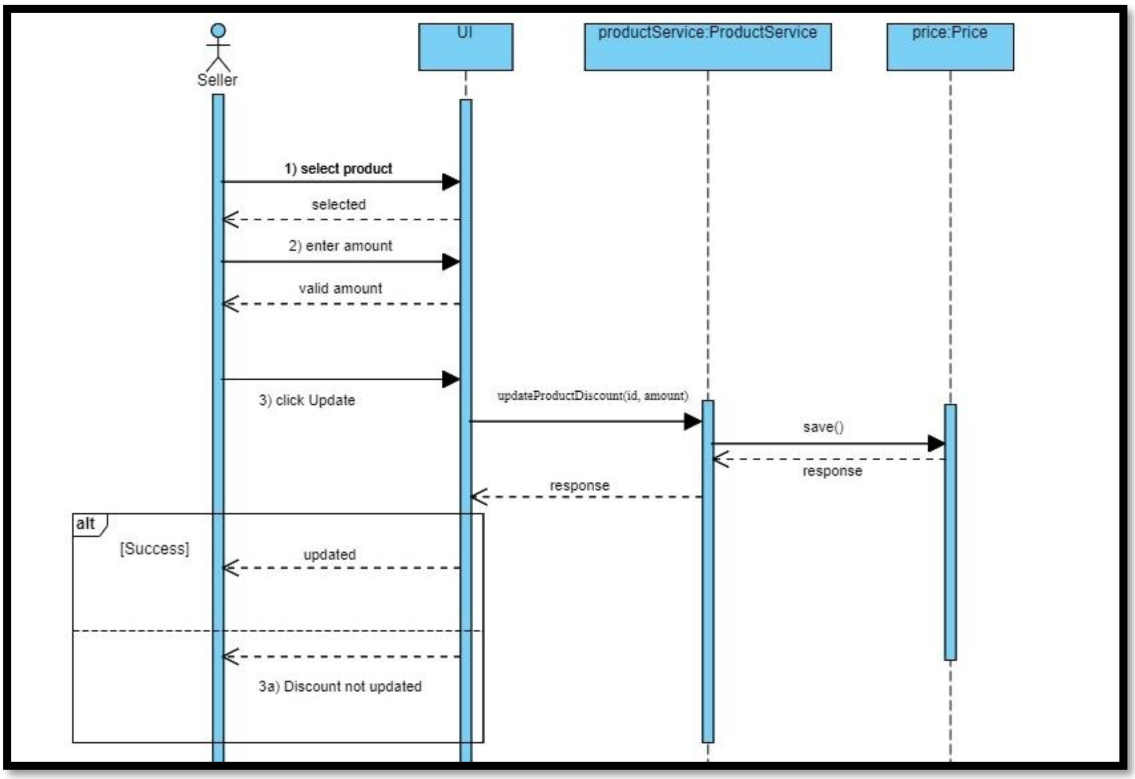


Figure 4:22 Update Product Discount

4.4.3. Customer Sequence

4.4.3.1. Login

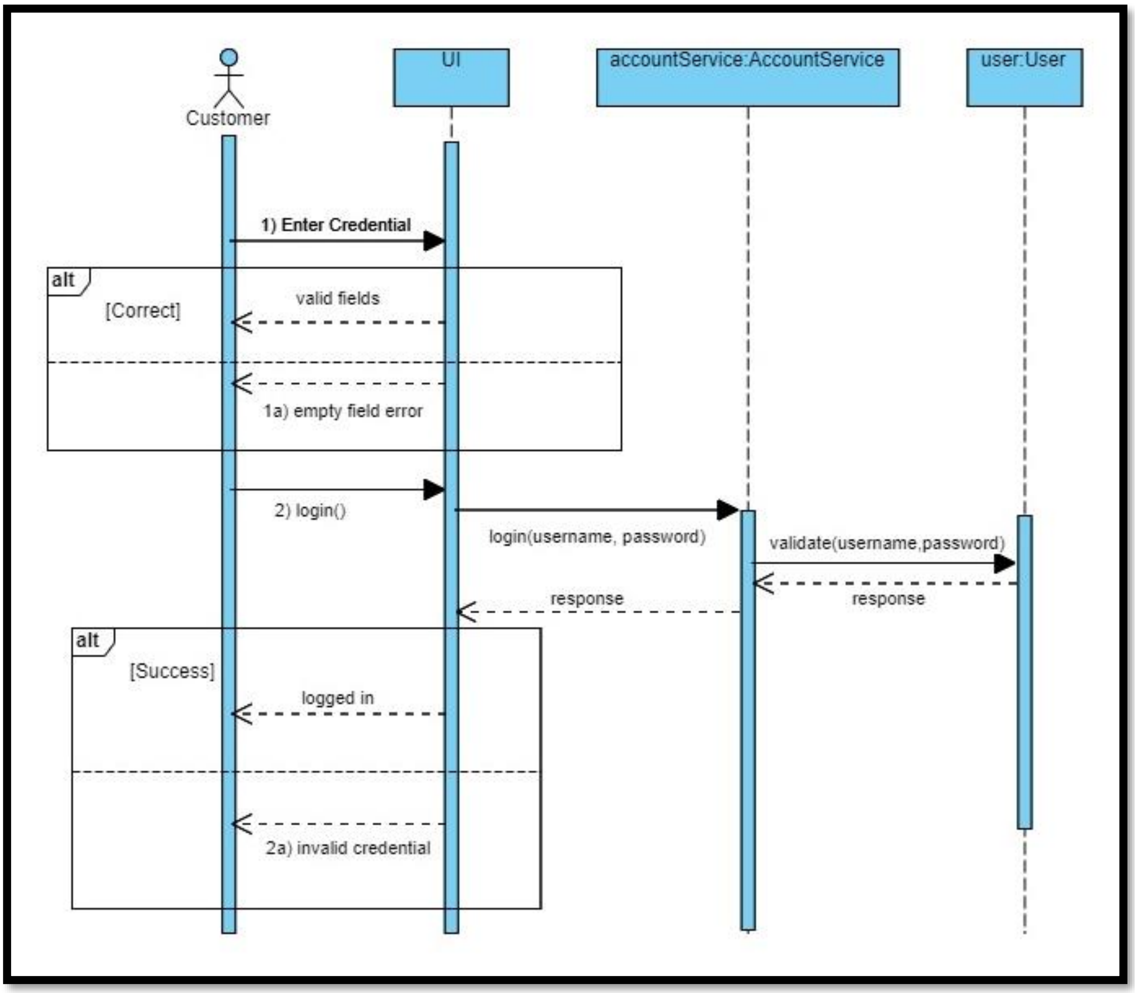


Figure 4:23 Customer Login

4.4.3.2. Register

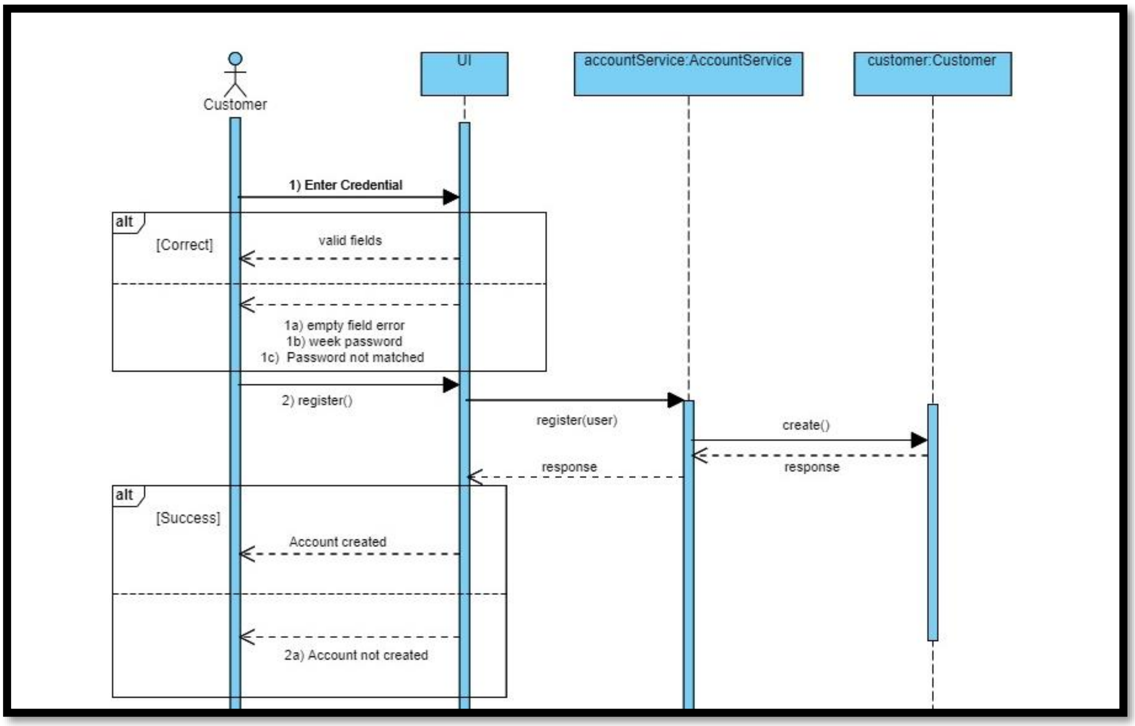


Figure 4:24 Customer Register

4.4.3.3. Recover Password

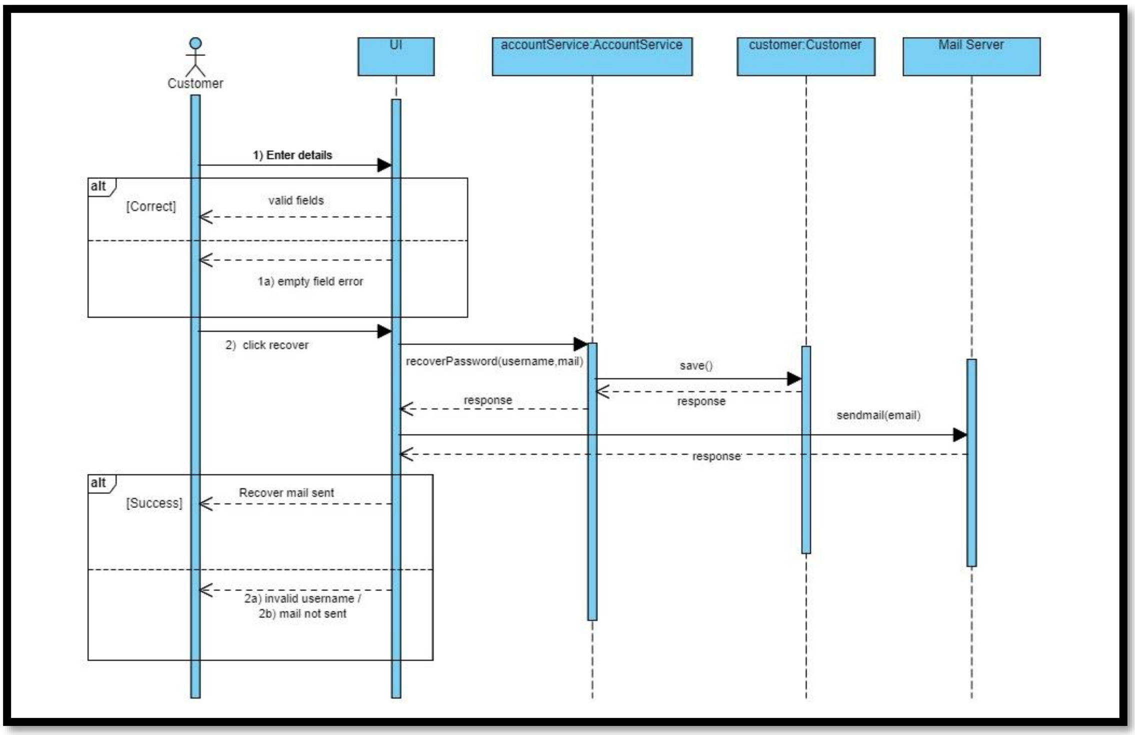


Figure 4:25 Recover Password

4.4.3.4. Update Profile

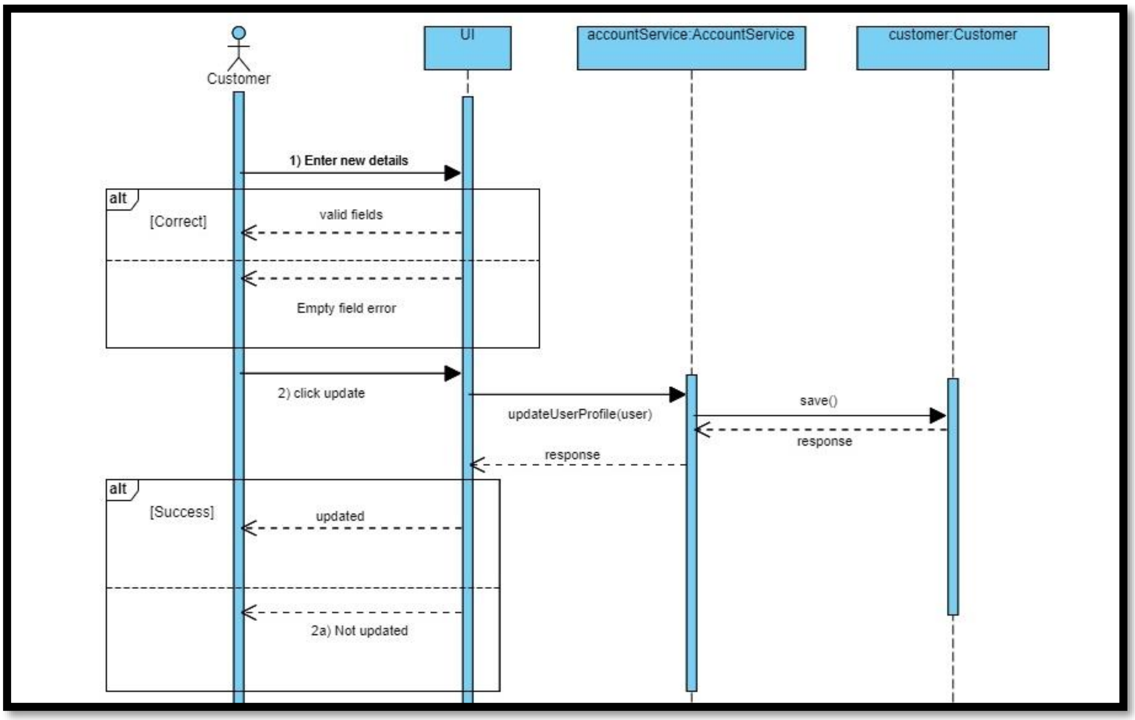


Figure 4:26 Update Profile

4.4.3.5. Search Product

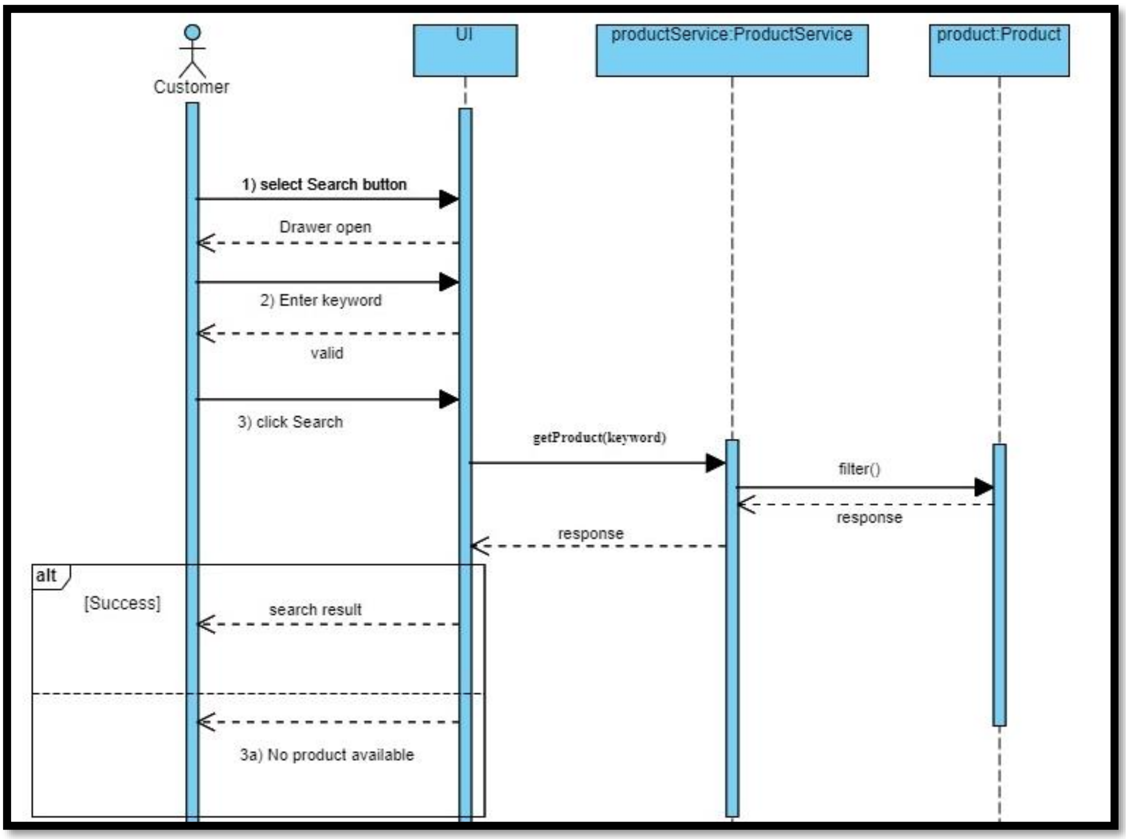


Figure 4:27 Search Product

4.4.3.6. Add To Cart

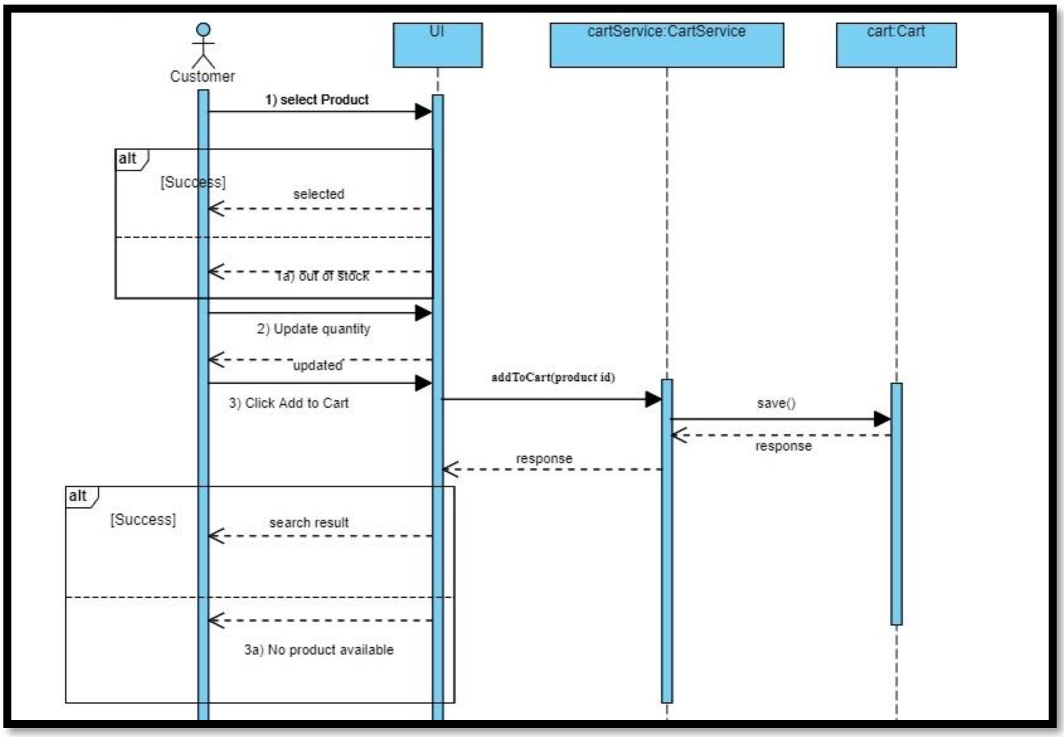


Figure 4:28 Add to Cart

4.4.3.7. Delete From Cart

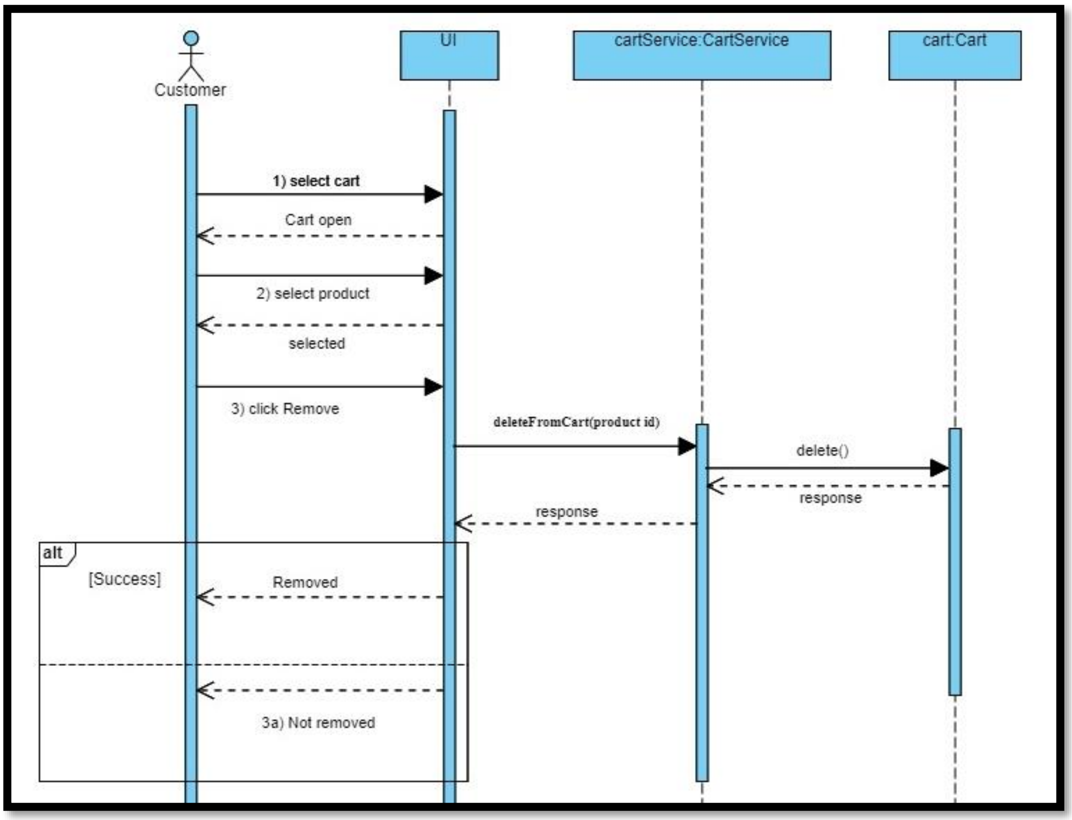


Figure 4:29 Delete from Cart

4.4.3.8. Checkout

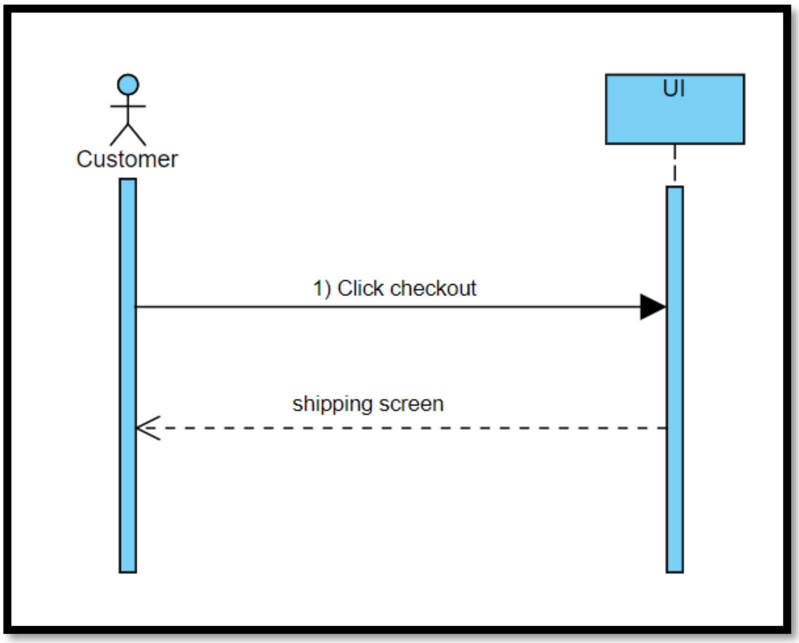


Figure 4:30 Checkout

4.4.3.9. Add Shipping detail

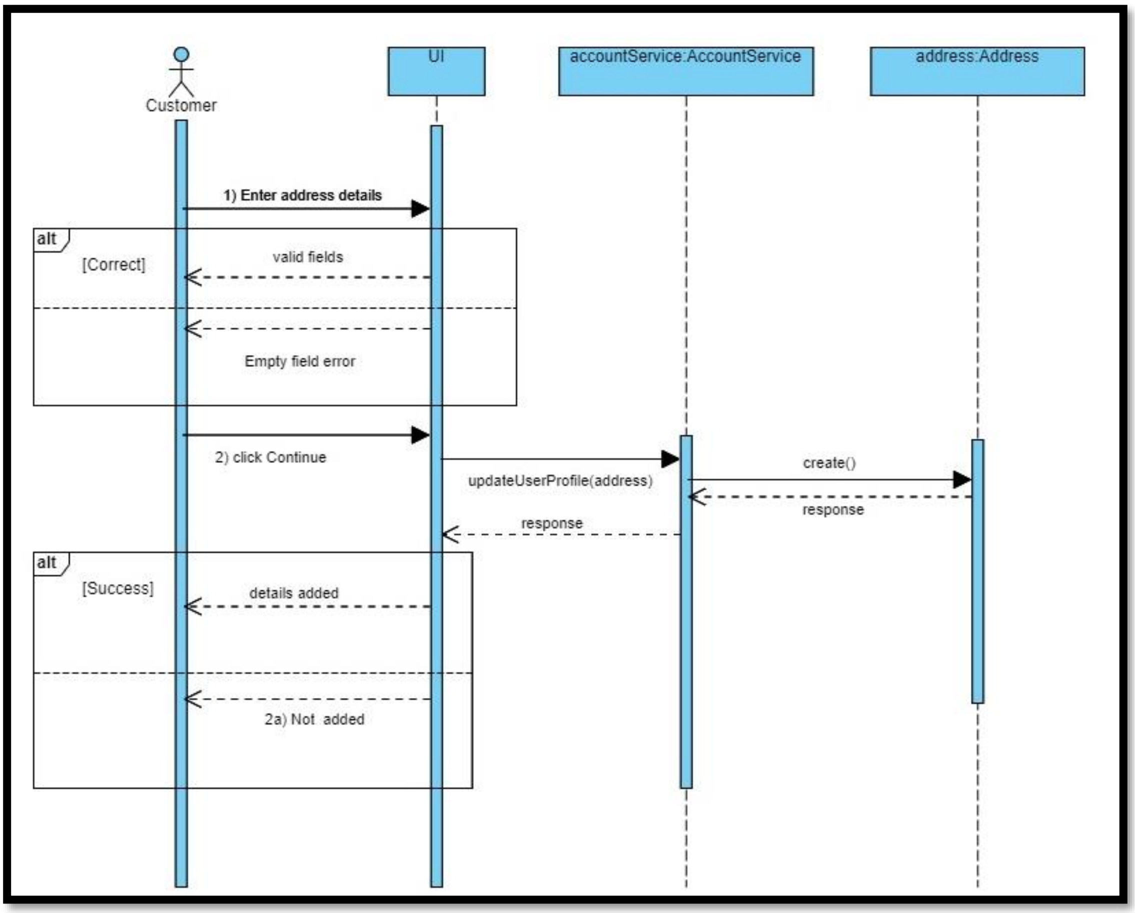


Figure 4:31 Add Shipping Detail

4.4.3.10. Select Payment Method

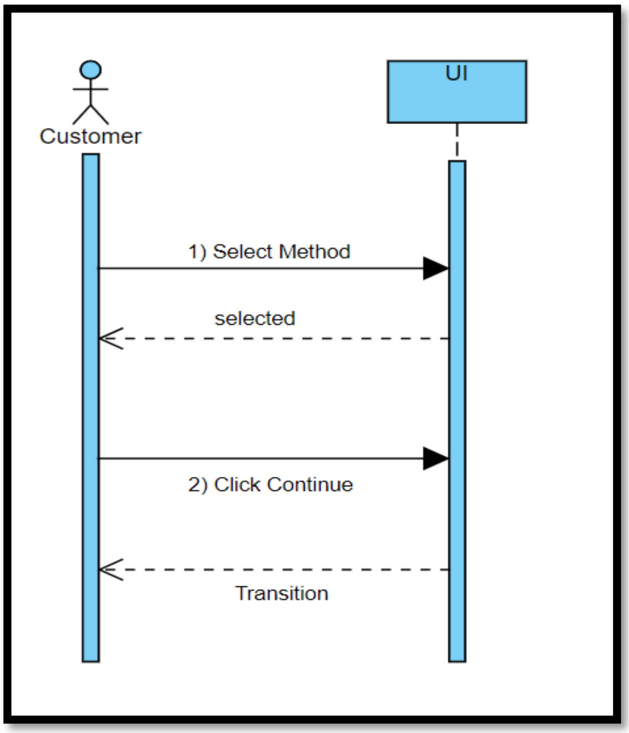


Figure 4:32 Select Payment Method

4.4.3.11. Place Order

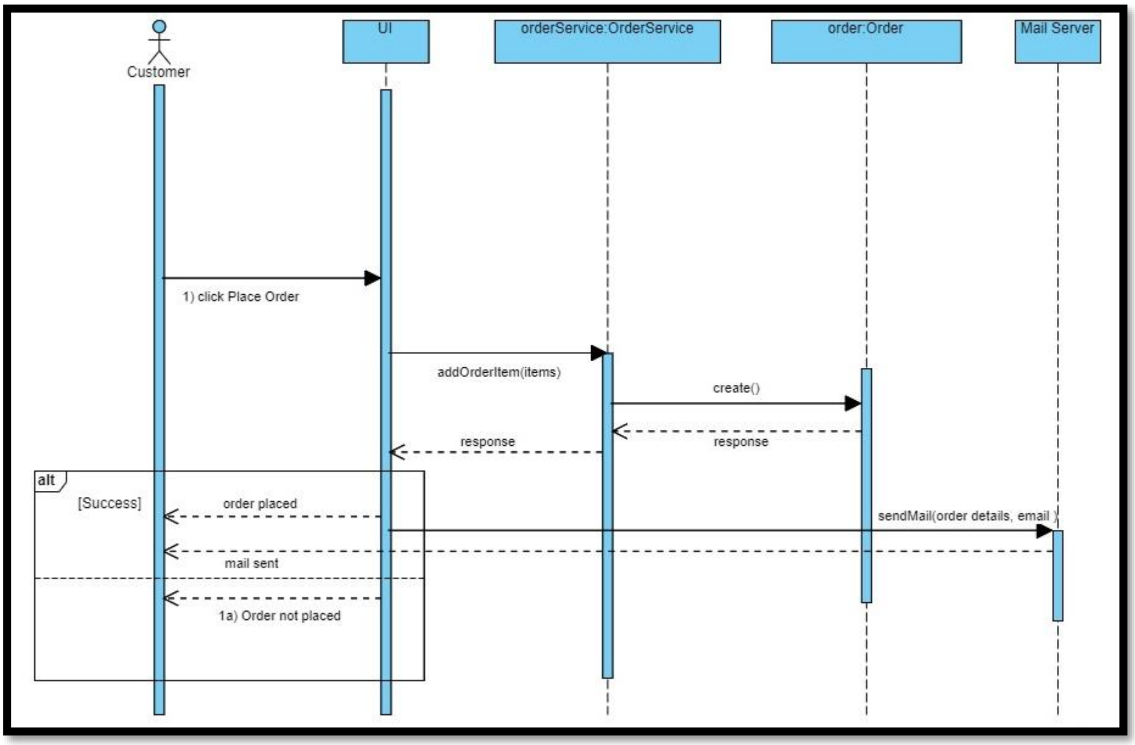


Figure 4:33 Place Order

4.4.3.12. Track Order

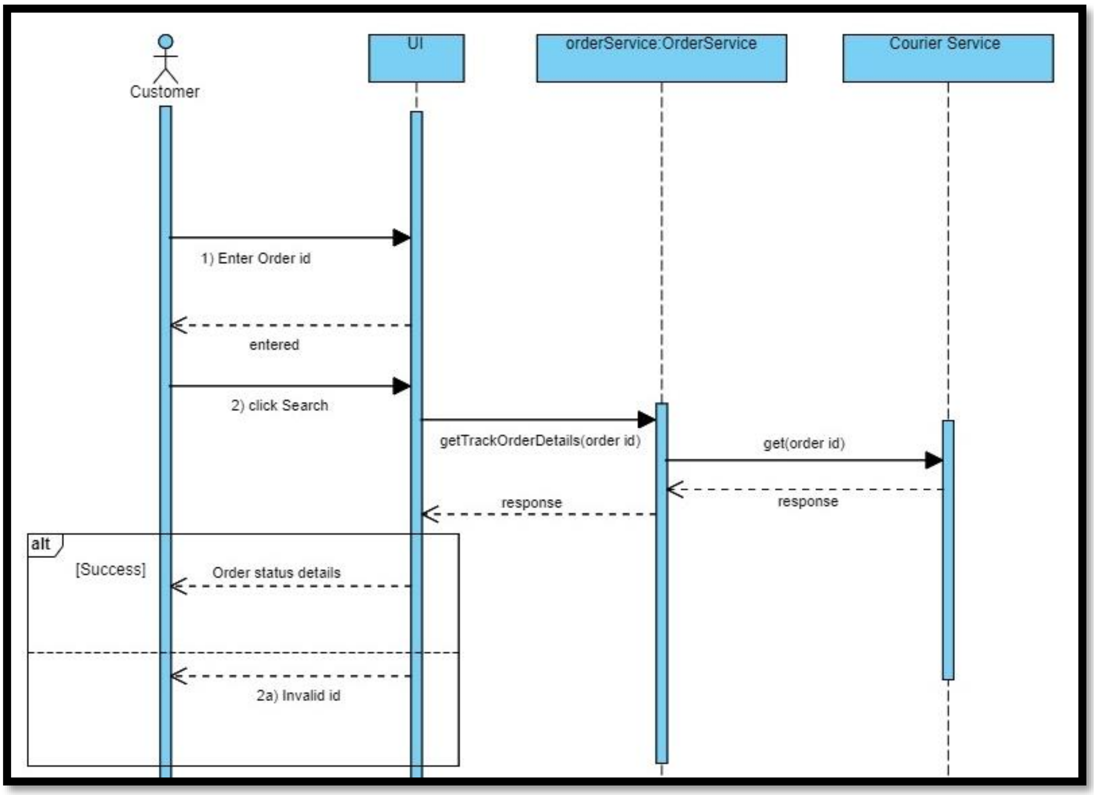


Figure 4:34 Track Order

4.4.3.13. Review Product

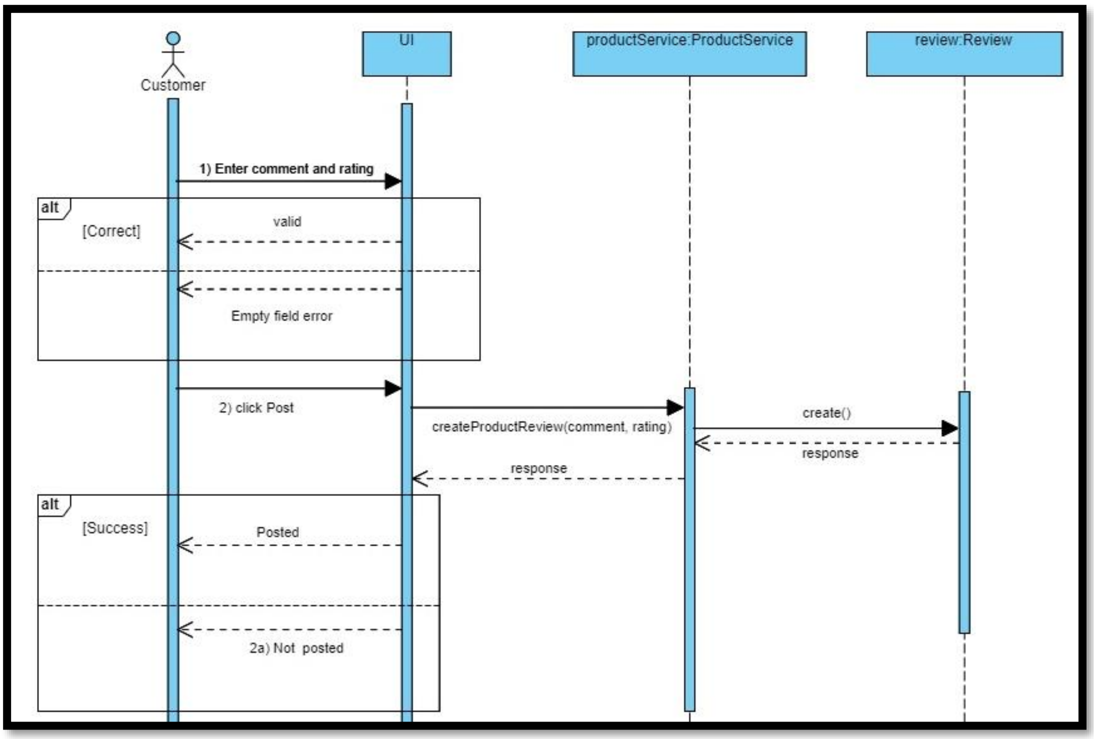


Figure 4:35 Review Product

4.4.3.14. Give Feedback

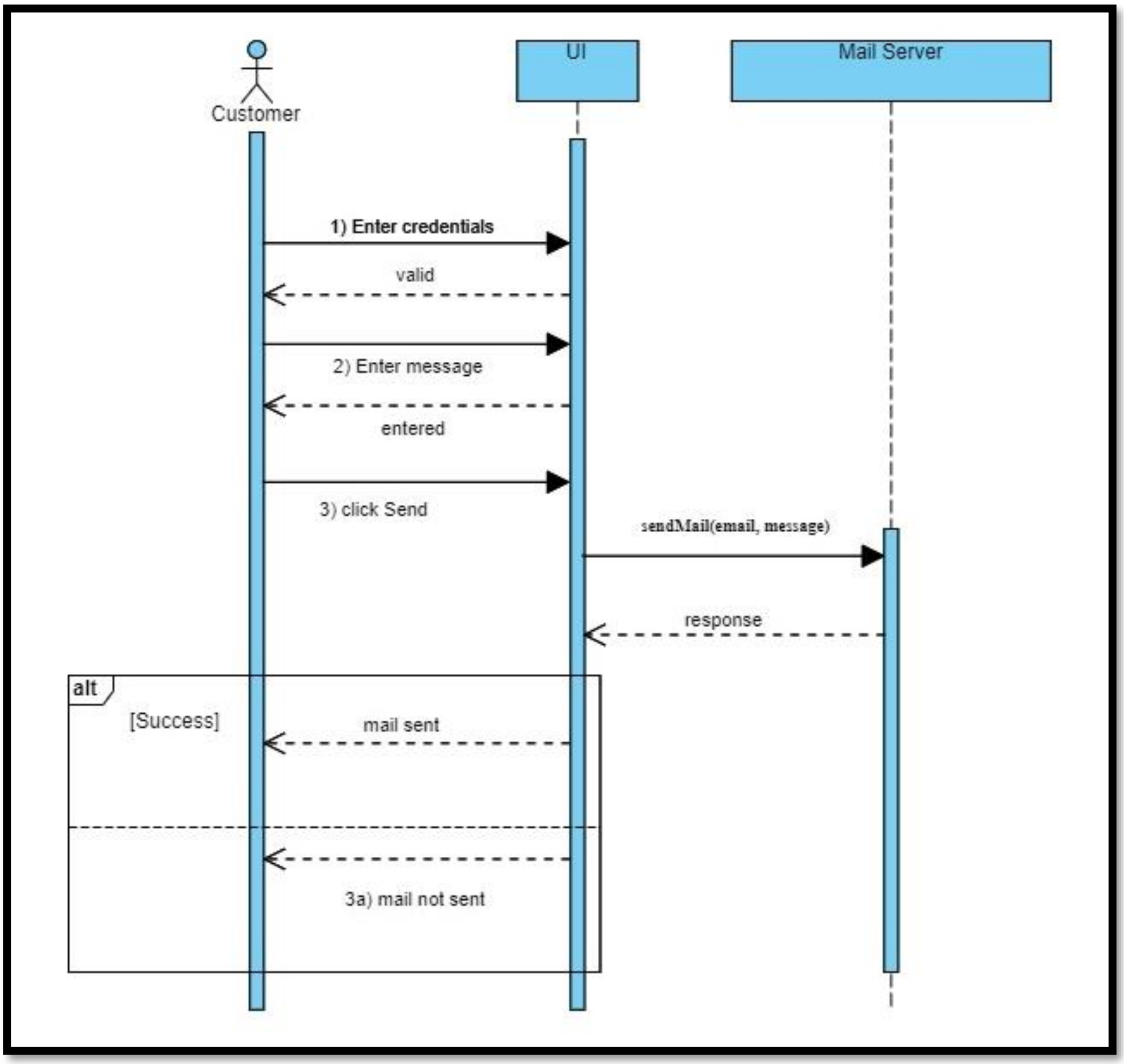


Figure 4:36 Give Feedback

4.4.3.15. Add Comment

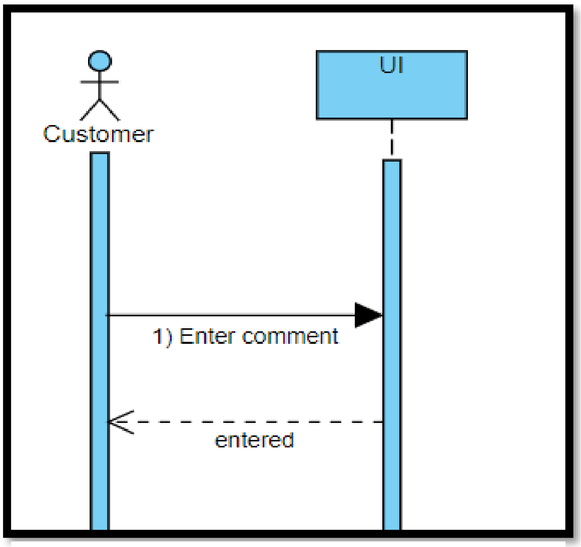


Figure 4:37 Add Comment

4.4.3.16. Apply Coupon

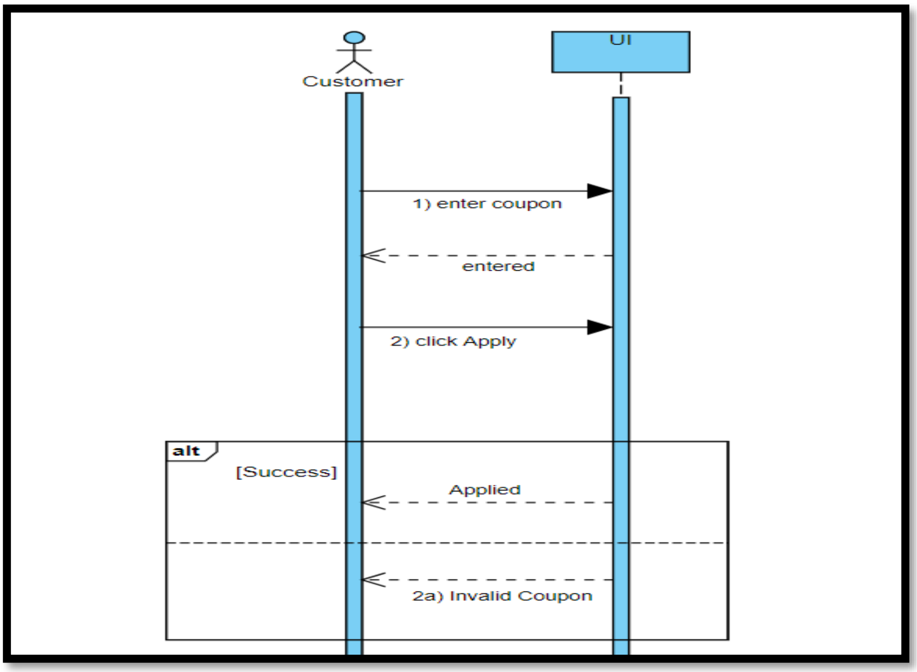


Figure 4:38 Apply Coupon

4.4.4. Admin Sequence Diagrams

4.4.4.1. Login

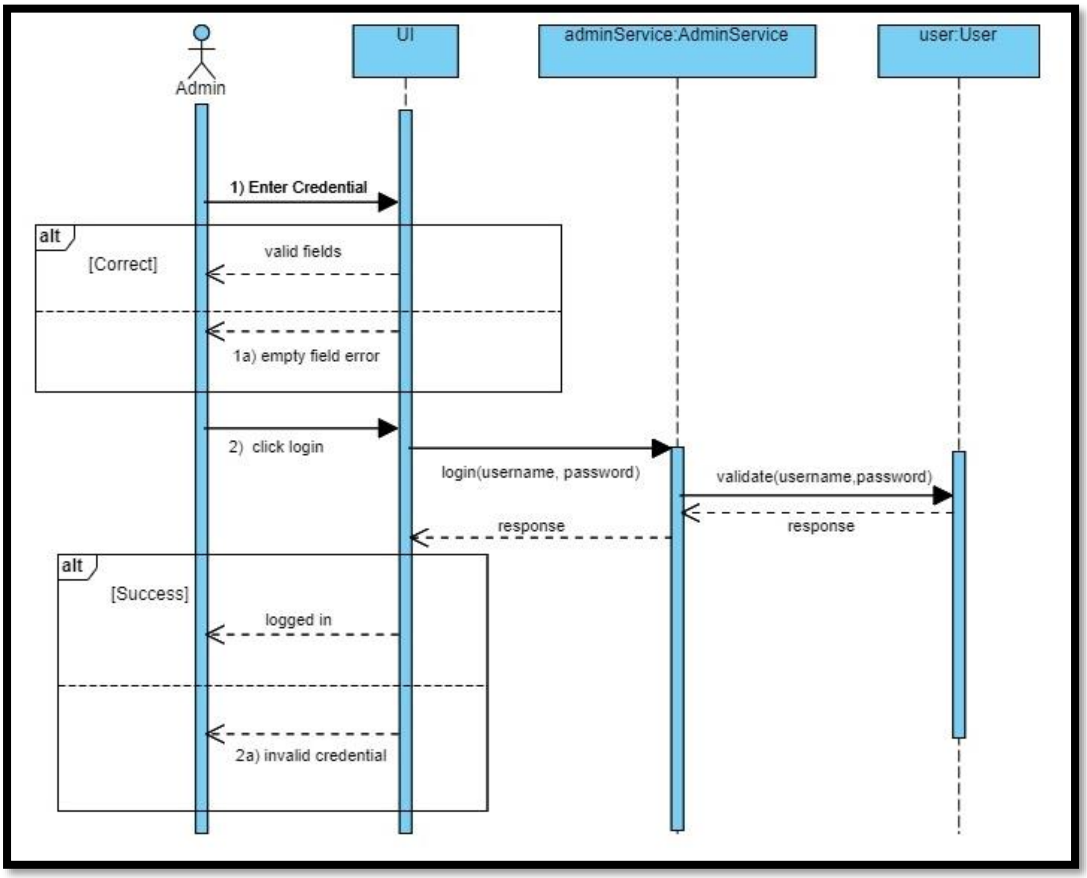


Figure 4:39 Admin Login

4.4.4.2. Verify Shop

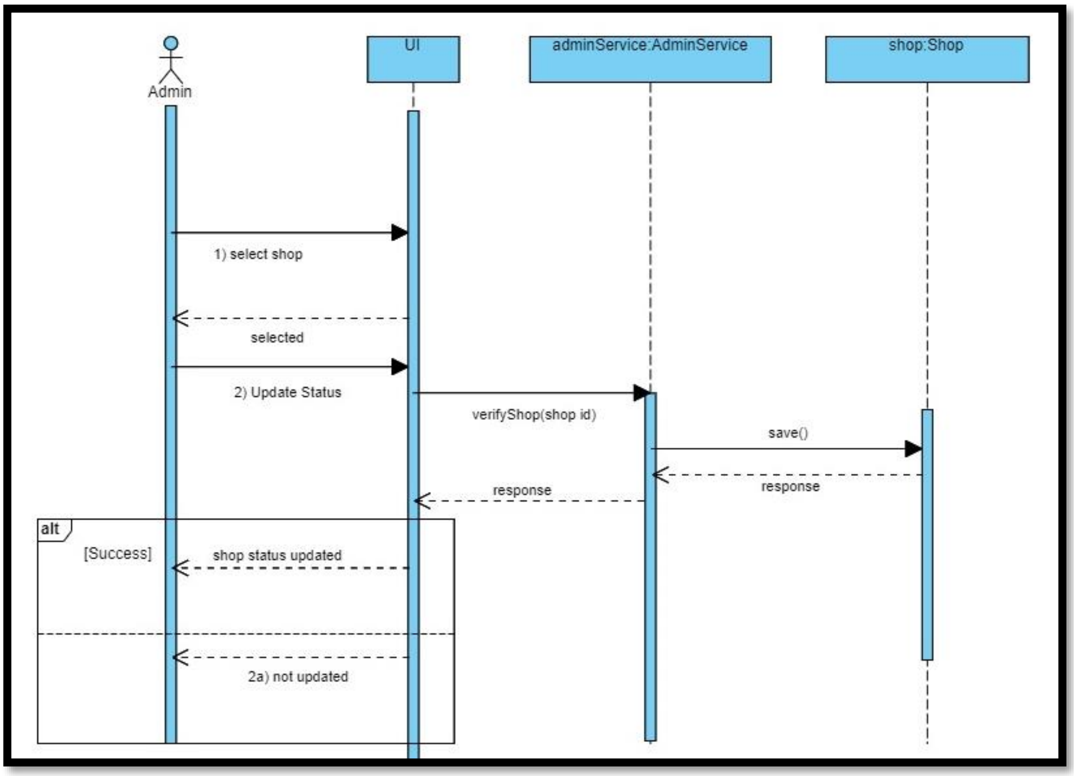


Figure 4:40 Verify Shop

4.4.4.3. Delete User

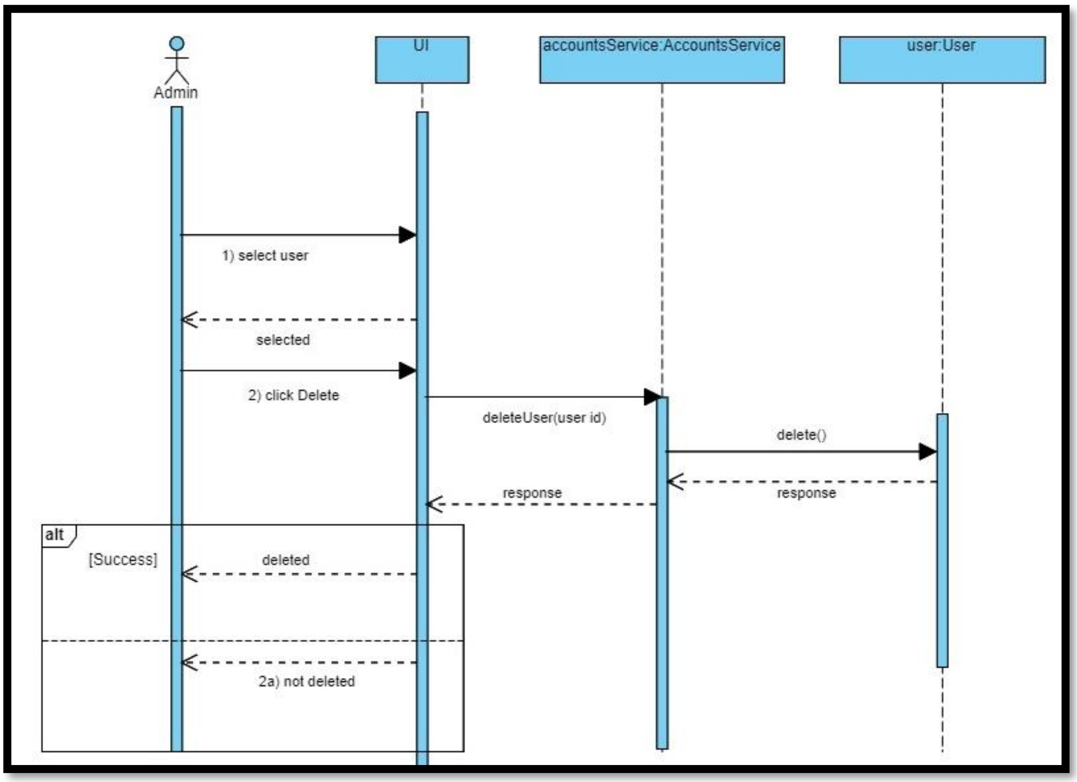


Figure 4:41 Delete User

4.4.4.4. Add Category

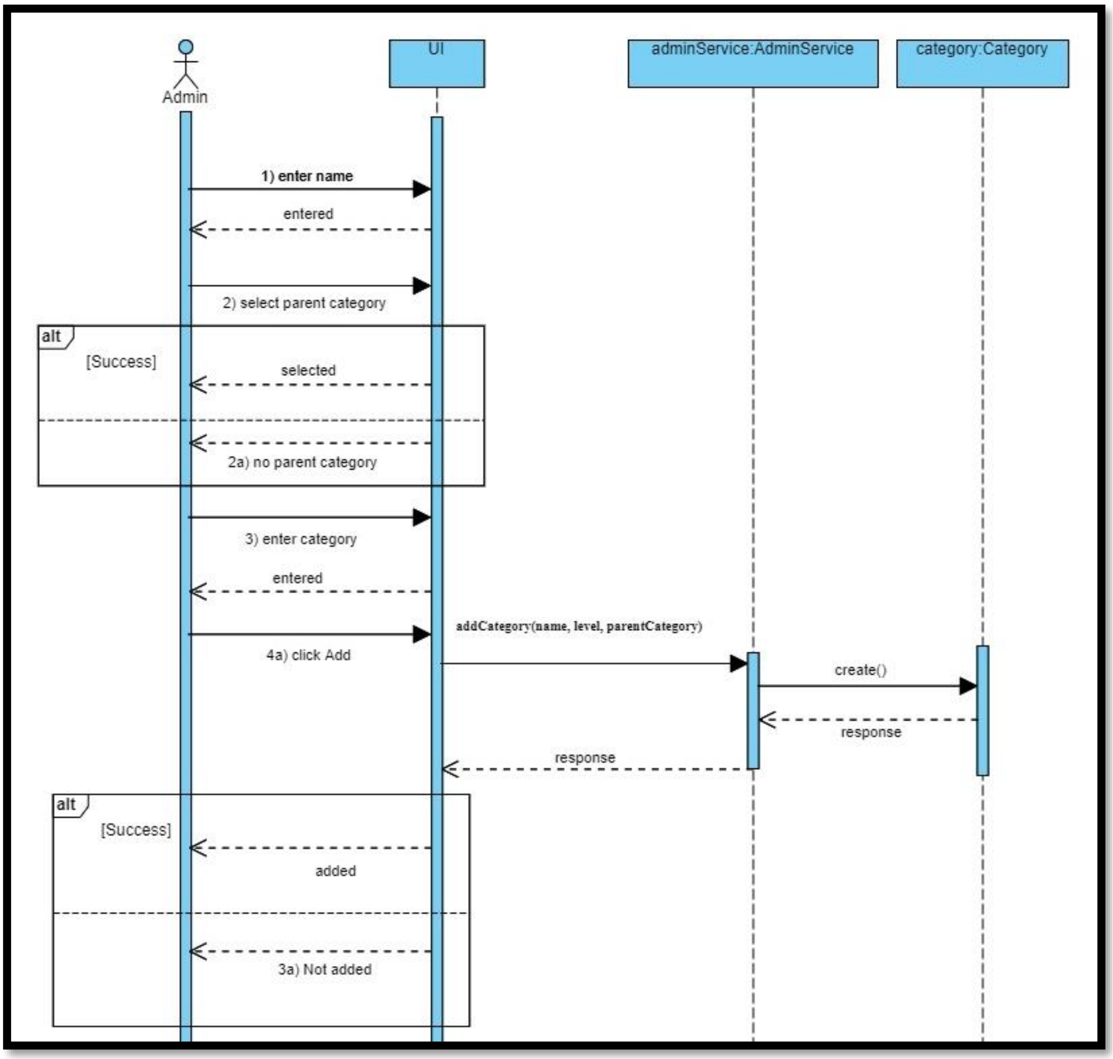


Figure 4:42 Add Category

4.4.4.5. Update Utils

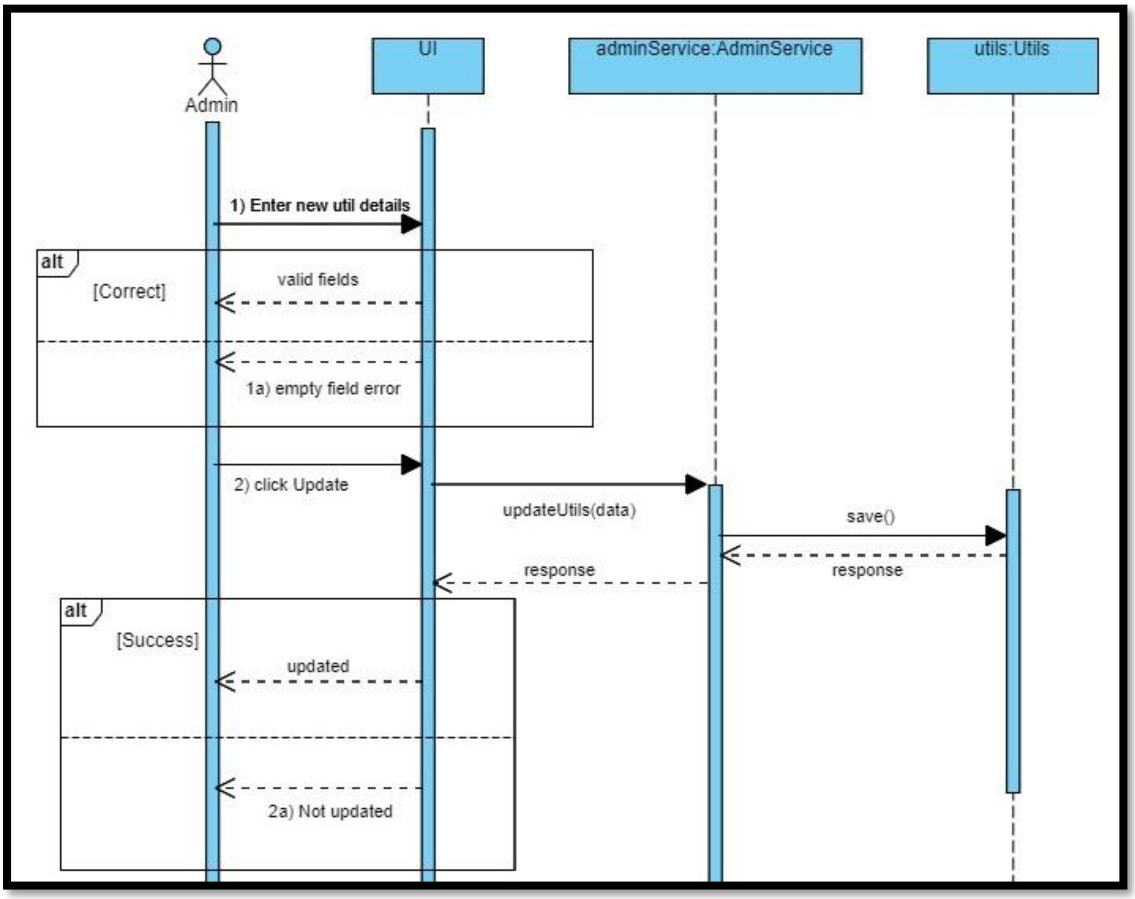


Figure 4:43 Update Utils

4.4.4.6. Update User Details

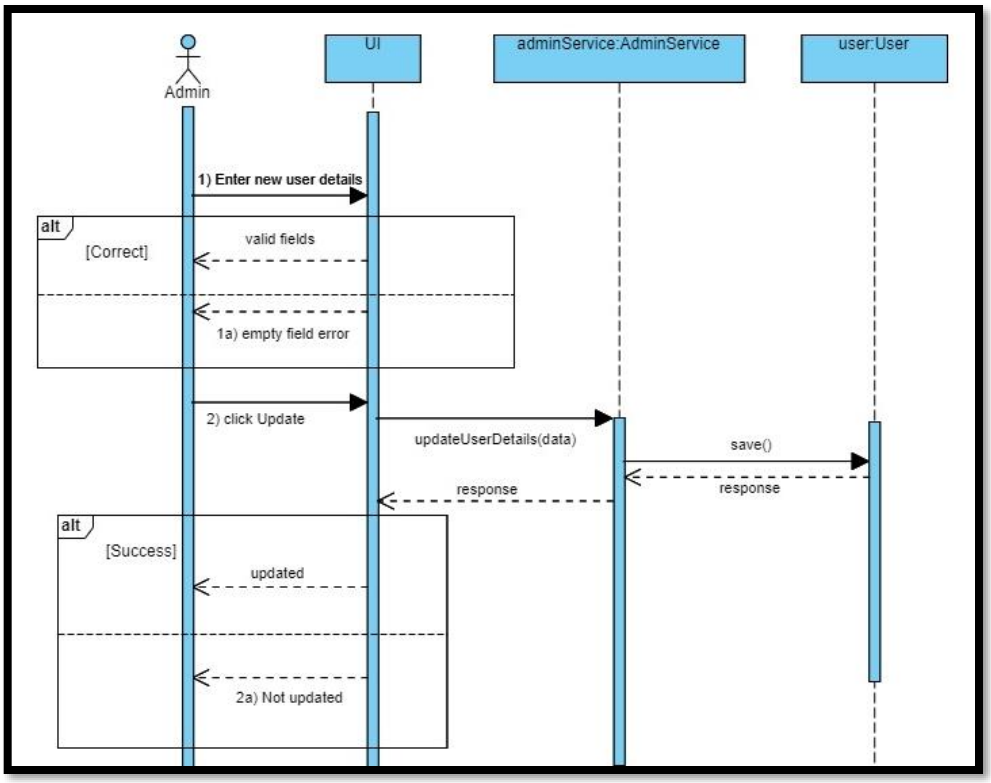


Figure 4:44 Update User Details

4.4.4.7. Update Order Status

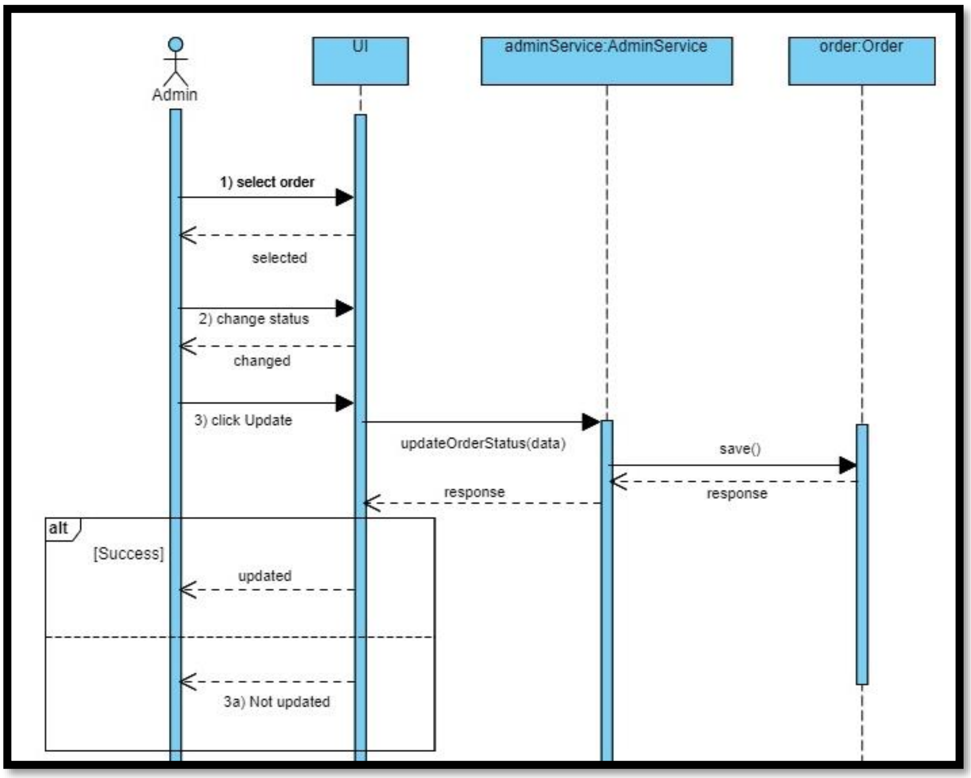


Figure 4:45 Update Order Status

4.4.4.8. Add Carousel Category

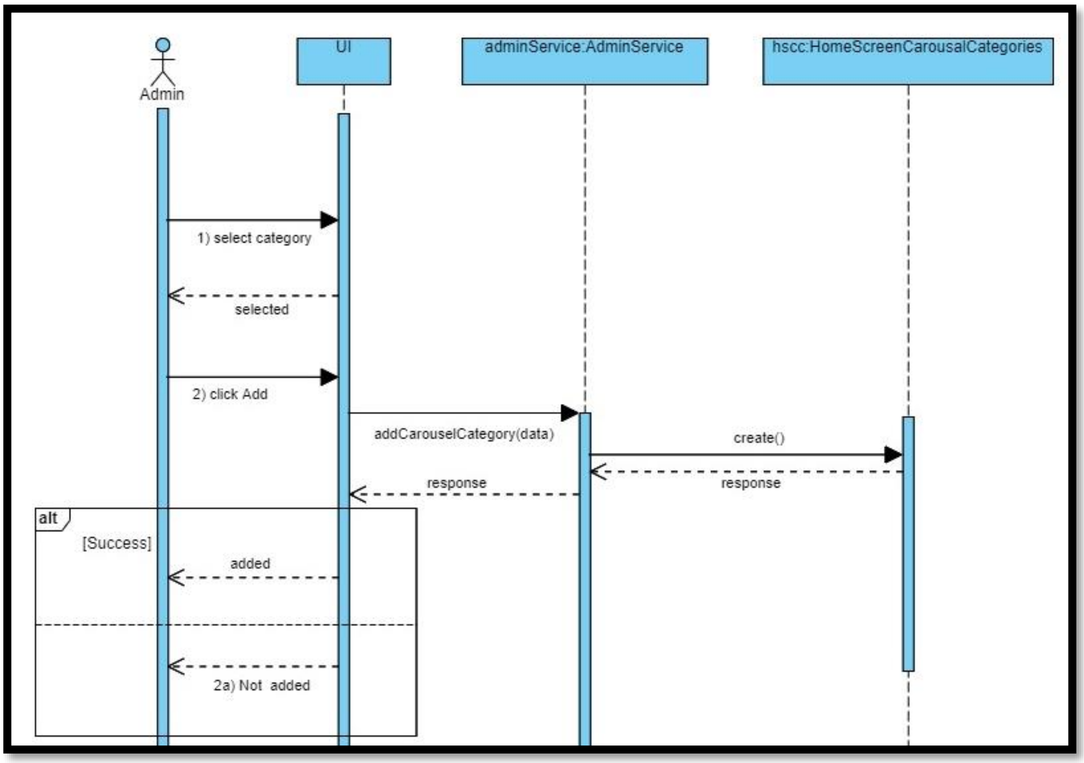


Figure 4:46 Add Carousal Category

4.4.4.9. Add Coupon

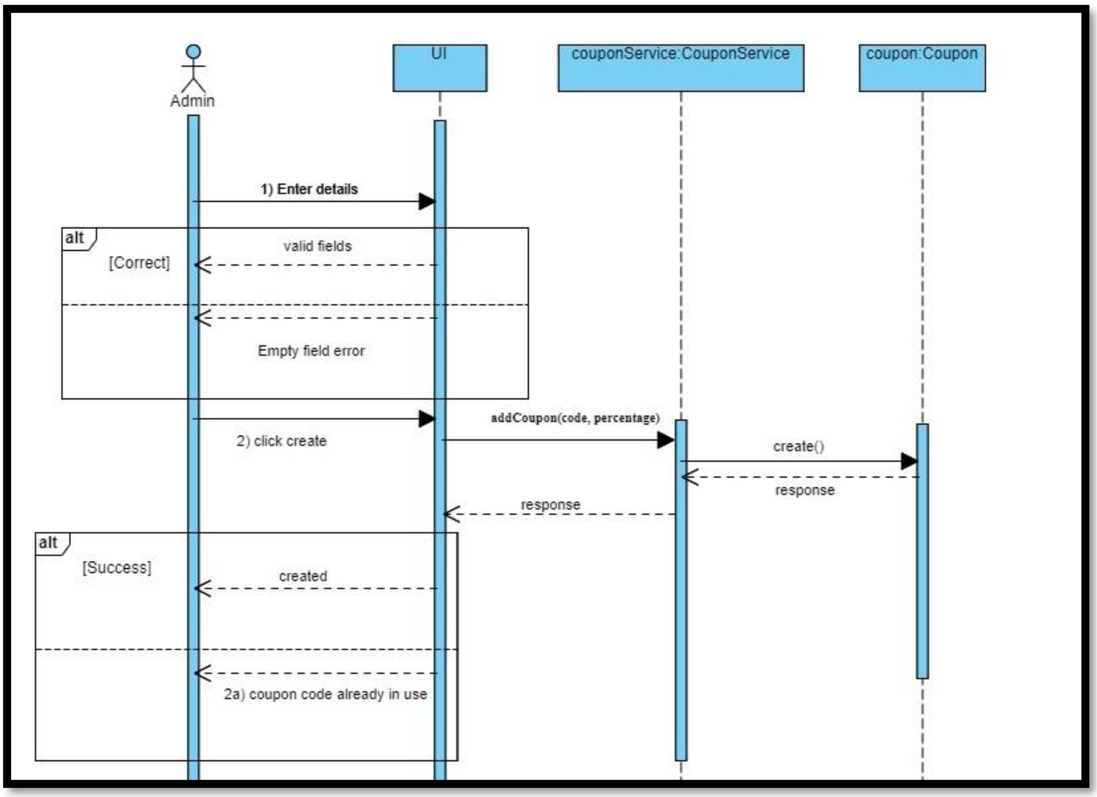


Figure 4:47 Add Coupon

4.4.4.10. Delete Coupon

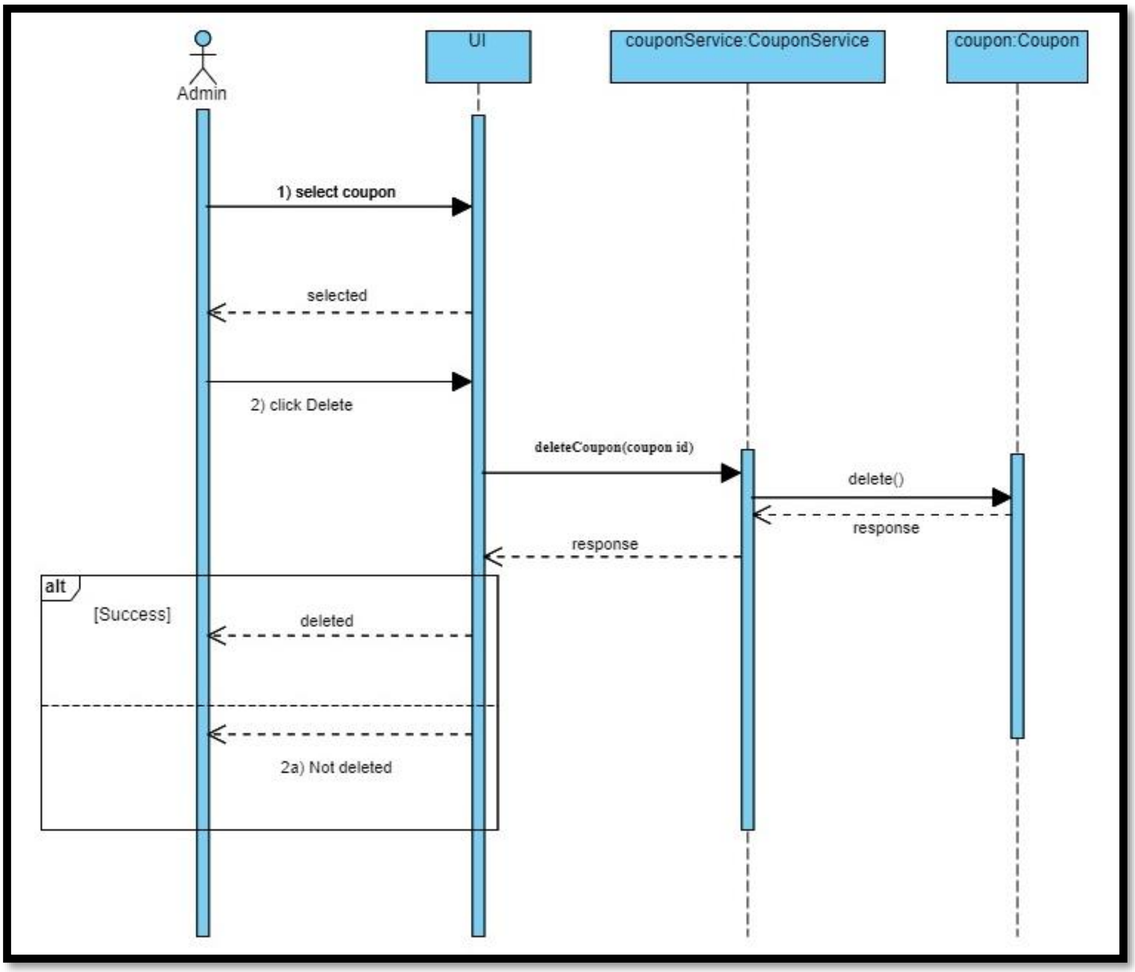


Figure 4:48 Delete Coupon

4.4.4.11. Update Coupon

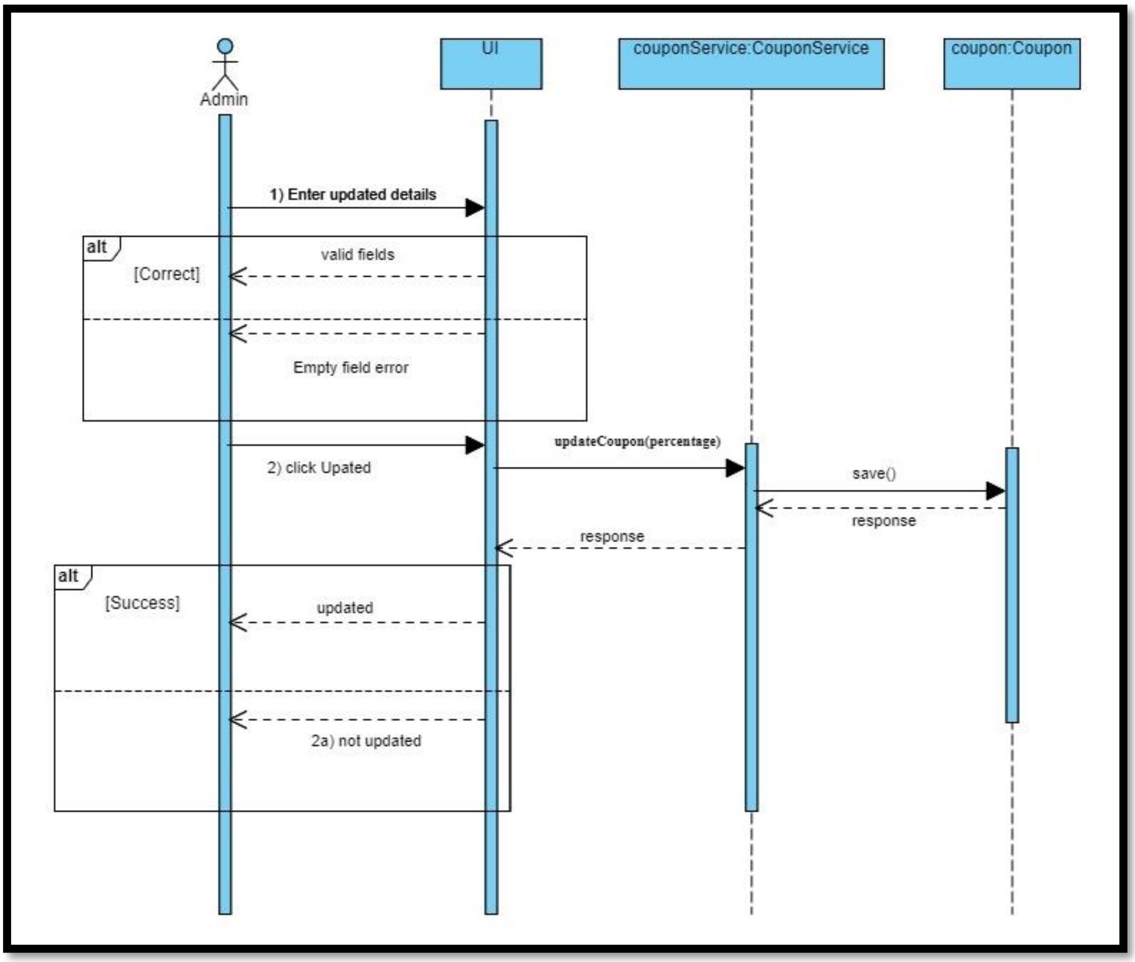


Figure 4:49 Update Coupon

4.4.5. Activity Diagrams

4.4.5.1. Seller Activity Diagram

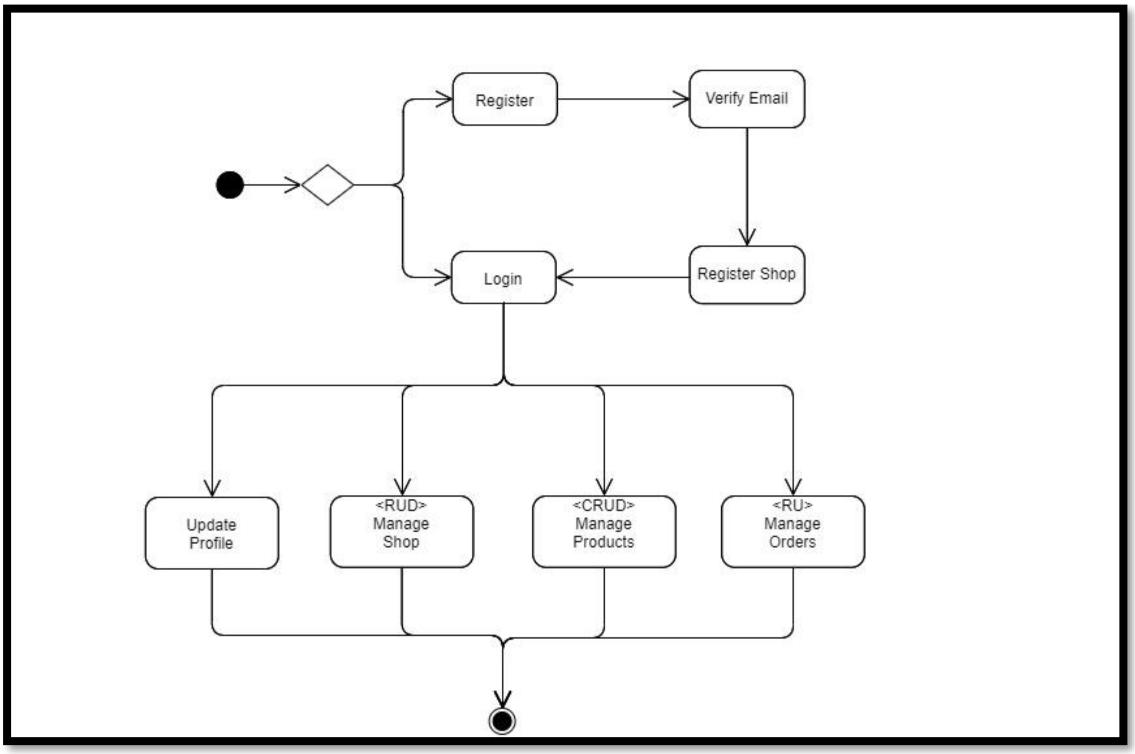


Figure 4:50 Seller Activity Diagram

4.4.5.2. Customer Activity Diagram

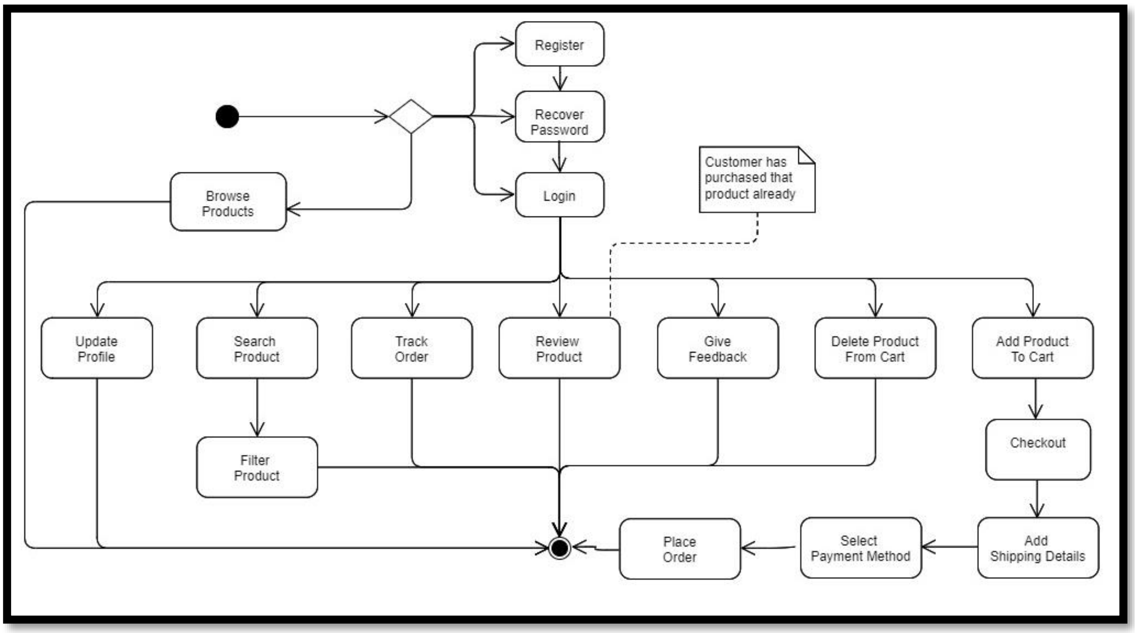


Figure 4:51 Customer Activity Diagram

4.4.5.3. Admin Activity Diagram

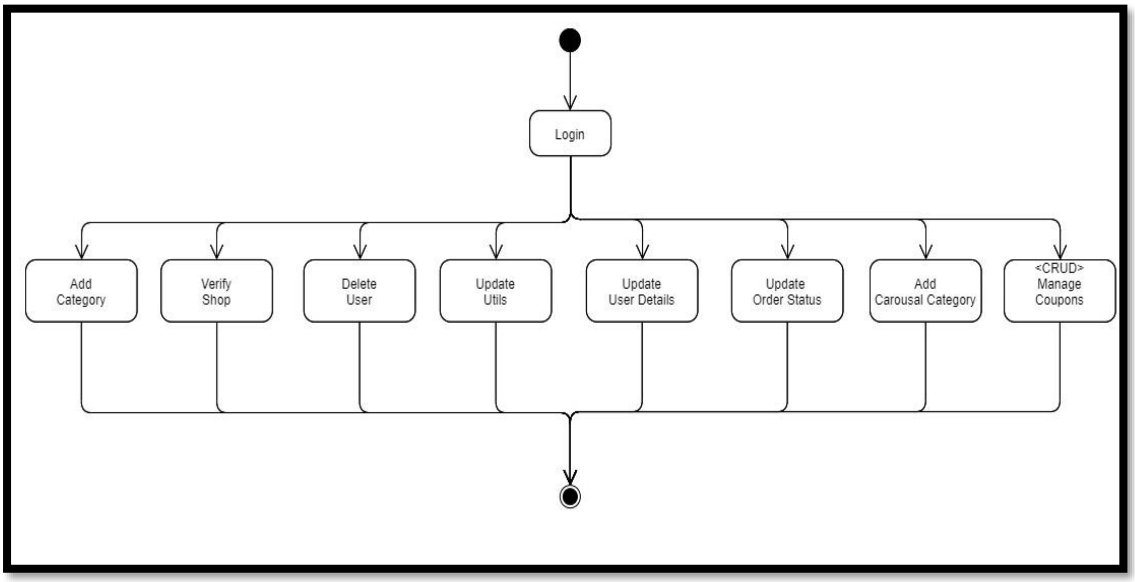


Figure 4:52 Admin Activity Diagram

4.5. Component Design

4.5.1. Deployment Diagram

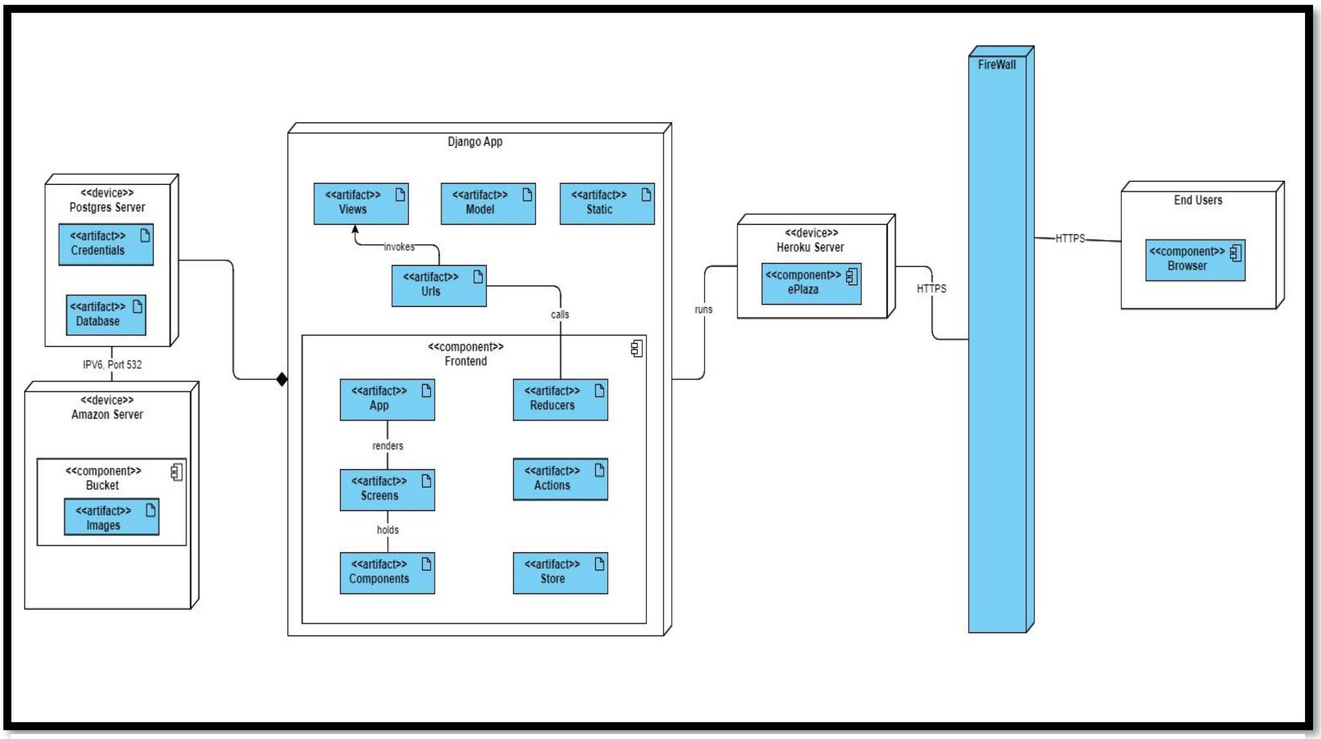


Figure 4:53 Deployment Diagram

4.6. Data Models

4.6.1. ER Diagram

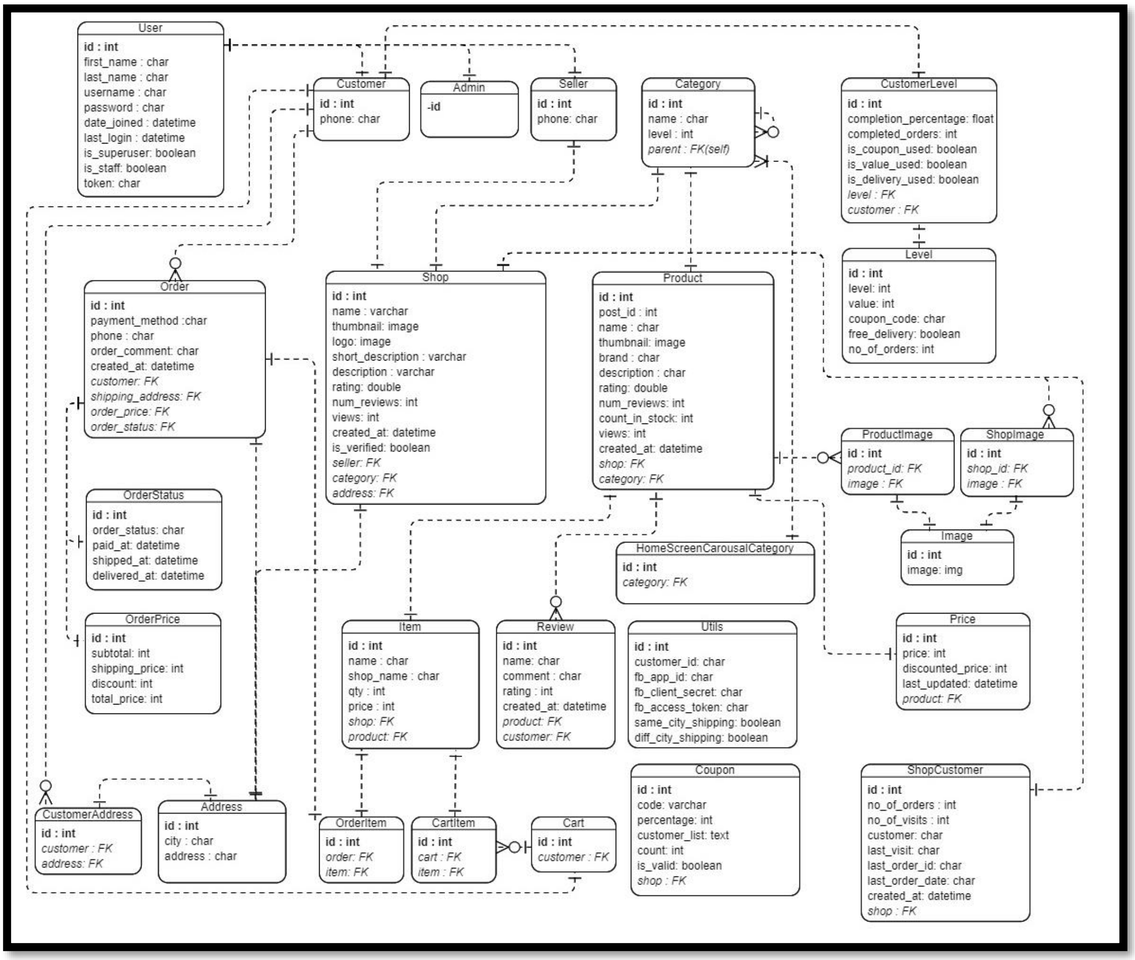


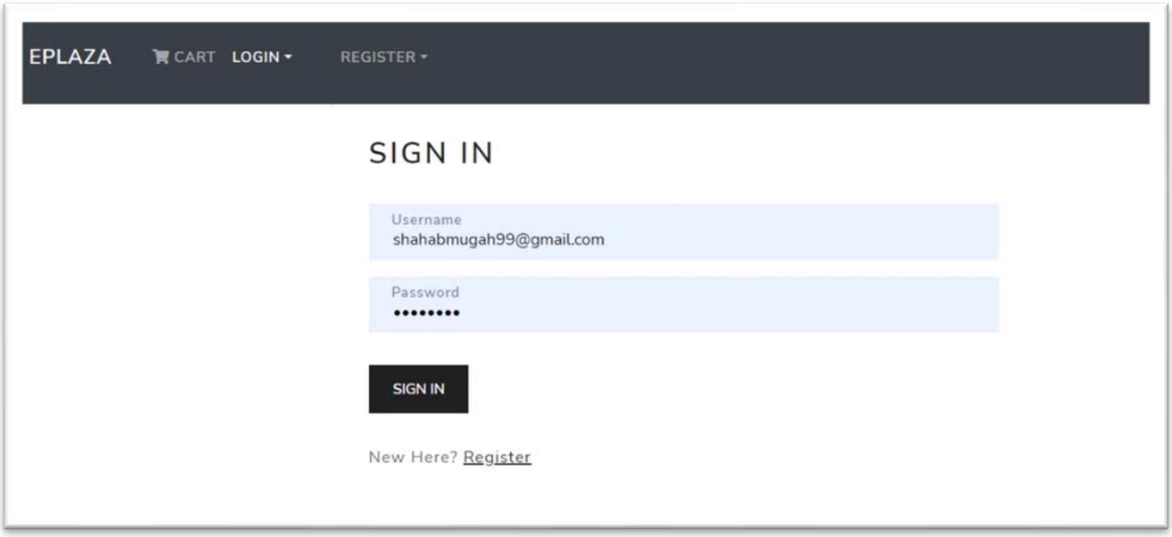
Figure 4:54 ER Diagram

4.7. System Prototype

The protype approach we used for our system is evolutionary, as it was decided in earlier development that the protype will be leading to final product by evolving product based on feedbacks.

Following are some early protype development Screenshots.

4.7.1. Login

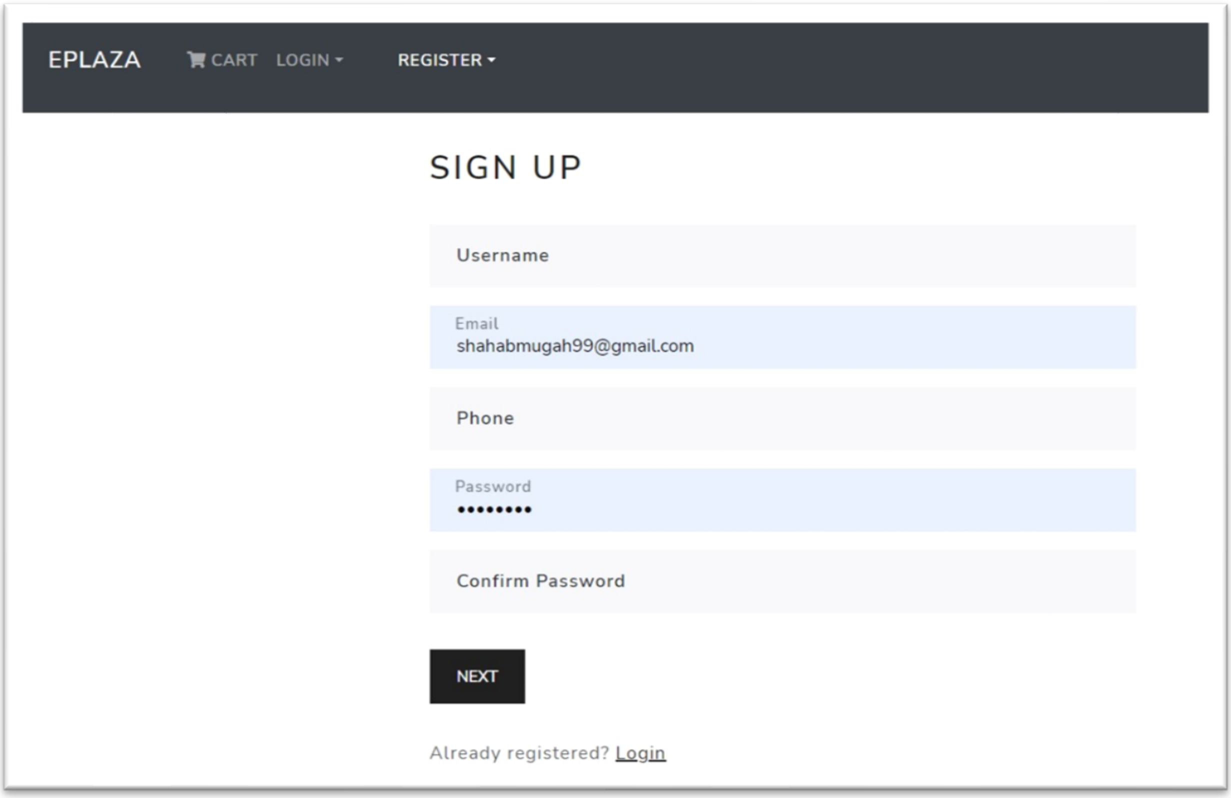


Username
shahabmugah99@gmail.com

Password
••••••••

Figure 4:55 Login Prototype

4.7.2. Register



Email
shahabmugah99@gmail.com

Password
••••••••

Figure 4:56 Register Prototype

4.7.3. Home

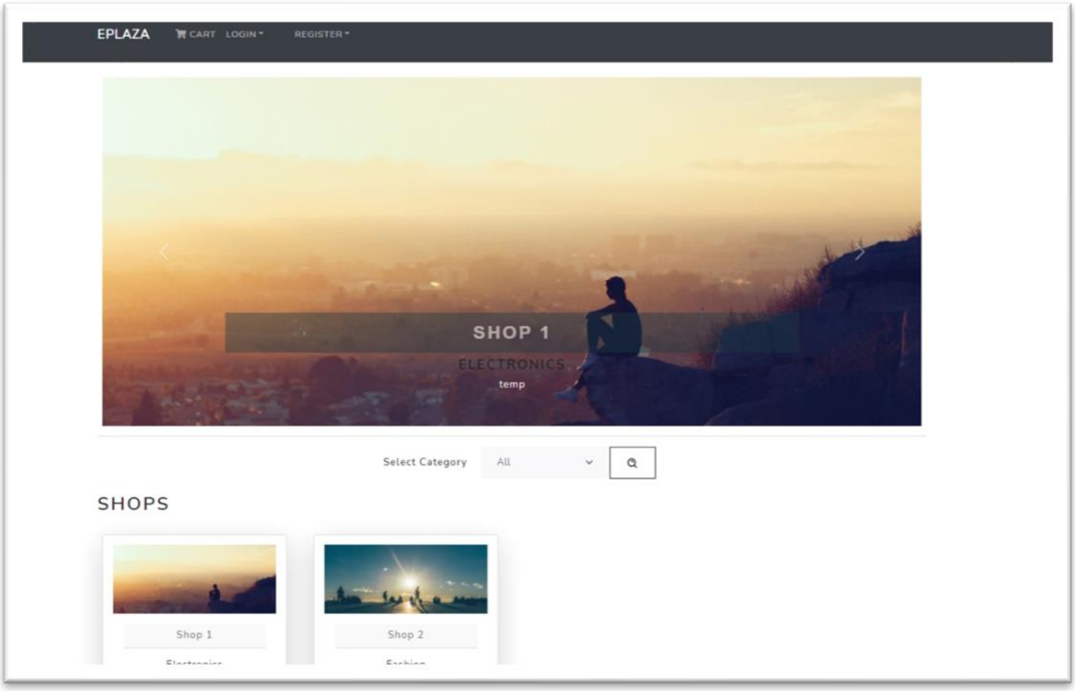


Figure 4:57 Home Prototype

4.7.4. Shop products

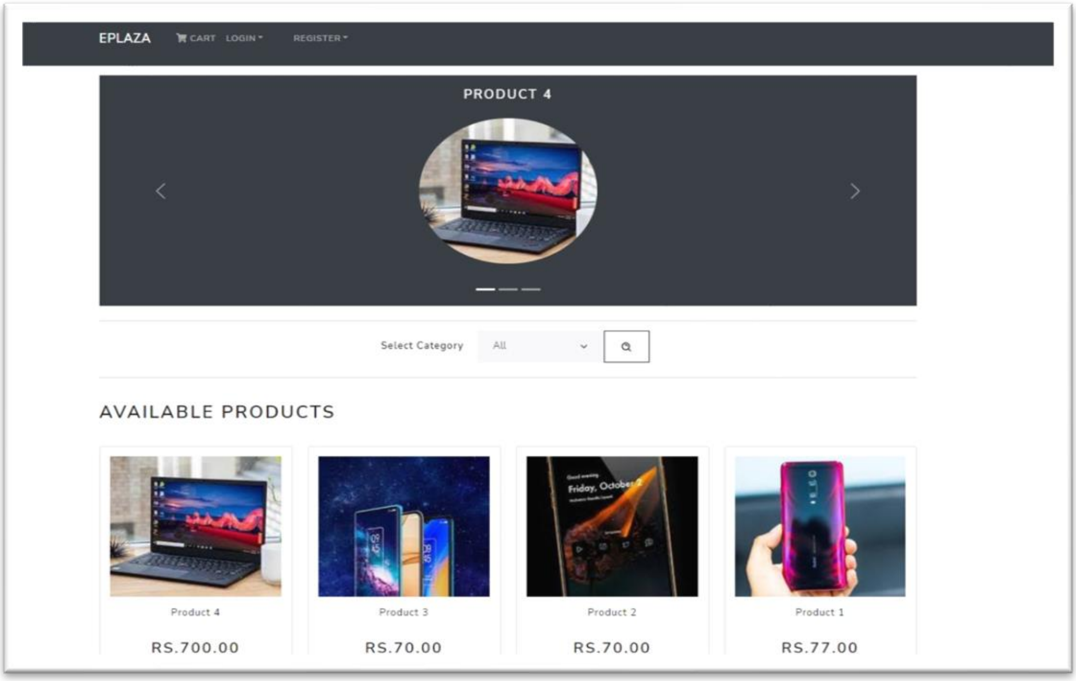
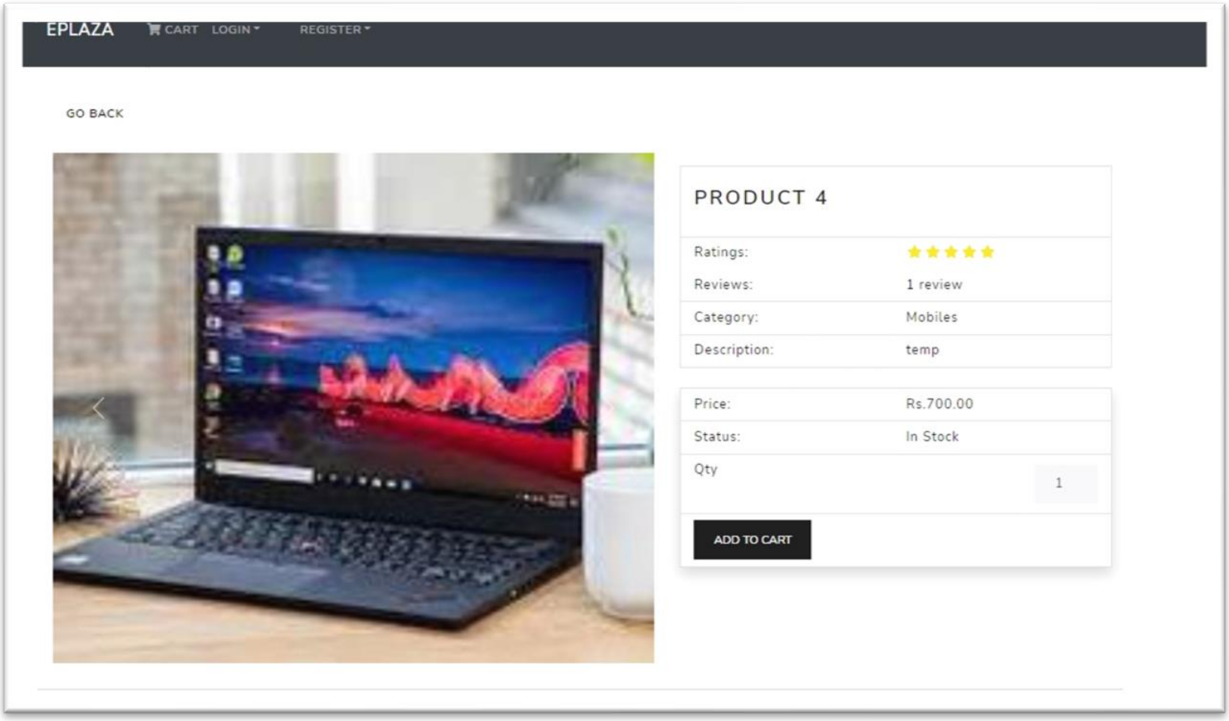


Figure 4:58 Shop Prototype

4.7.5. Product Detail

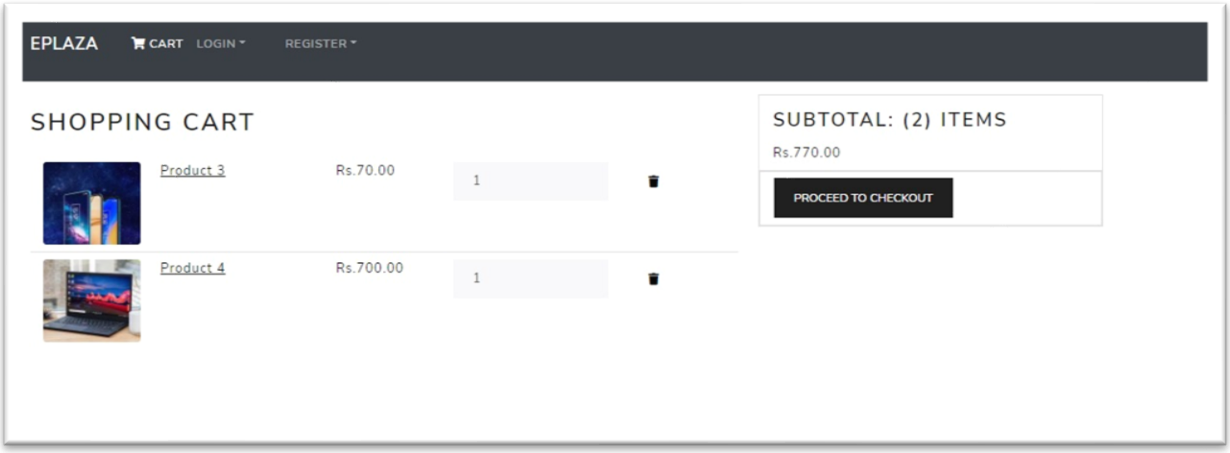


PRODUCT 4

Ratings:	★★★★★
Reviews:	1 review
Category:	Mobiles
Description:	temp
Price:	Rs.700.00
Status:	In Stock
Qty	1

Figure 4:59 Product Prototype

4.7.6. Cart



SHOPPING CART

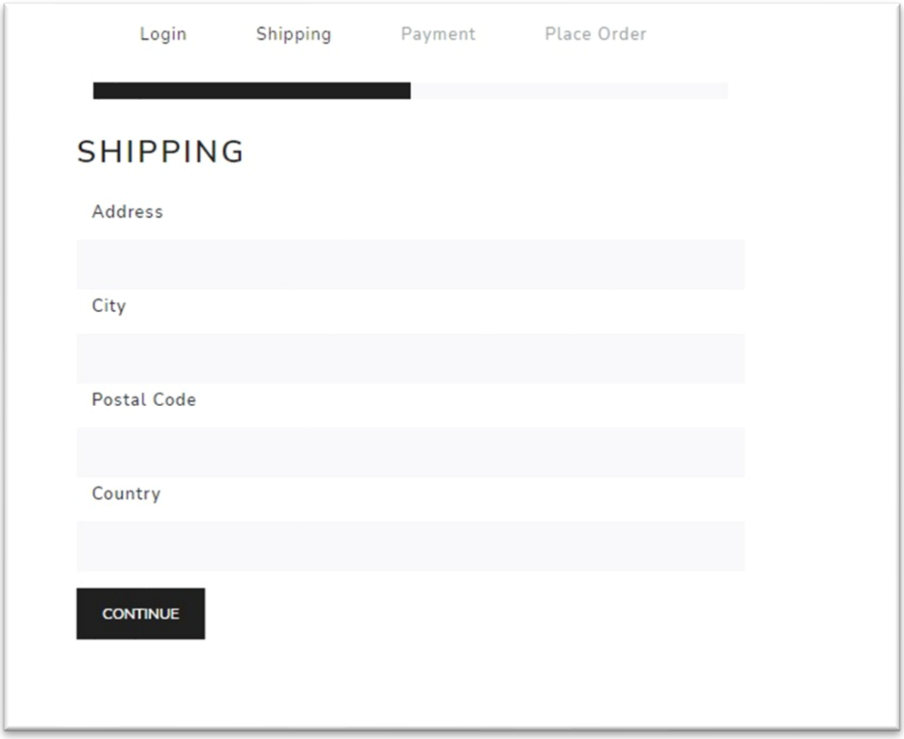
Product 3	Rs.70.00	1
Product 4	Rs.700.00	1

SUBTOTAL: (2) ITEMS

Rs.770.00

Figure 4:60 Cart Prototype

4.7.7. Shipping Detail



SHIPPING

Address

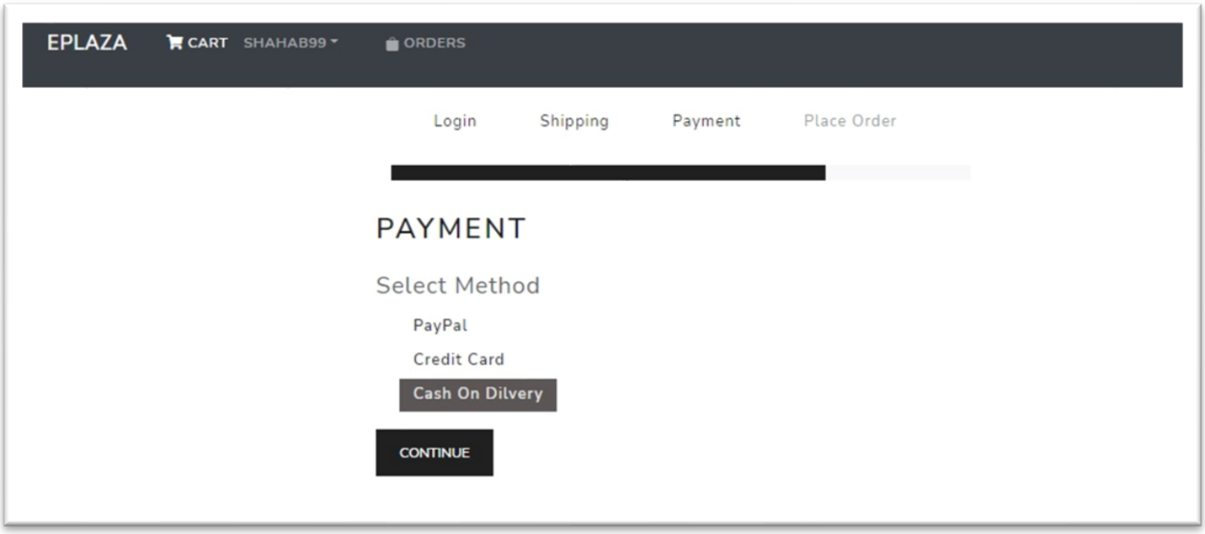
City

Postal Code

Country

Figure 4:61 Shipping Prototype

4.7.8. Payment Method



PAYMENT

Select Method

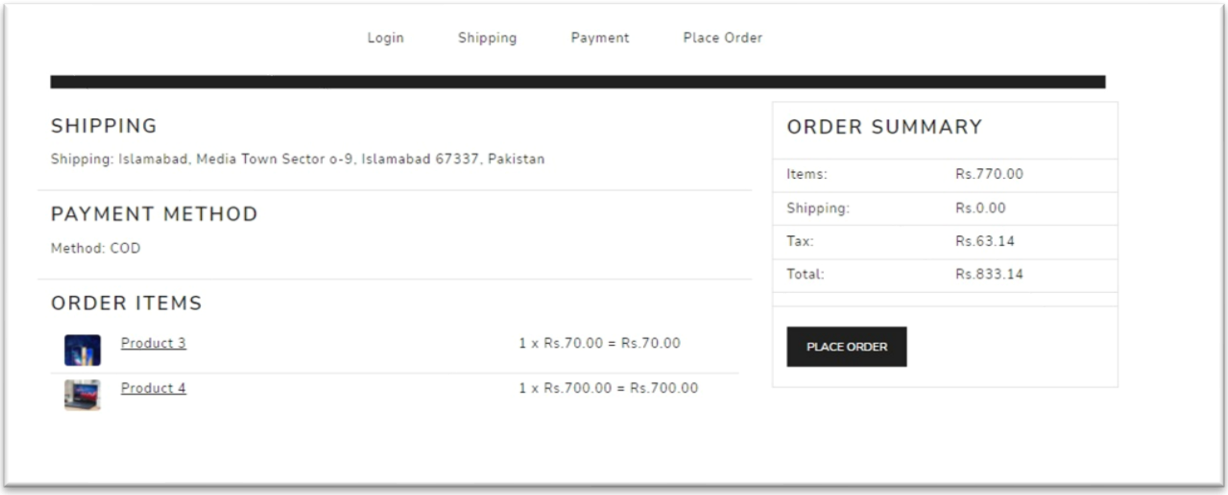
PayPal

Credit Card

Cash On Dilvery

Figure 4:62 Payment Method Protype

4.7.9. Place Order



Login Shipping Payment Place Order

SHIPPING

Shipping: Islamabad, Media Town Sector o-9, Islamabad 67337, Pakistan

PAYMENT METHOD

Method: COD

ORDER ITEMS

Product 3	1 x Rs.70.00 = Rs.70.00
Product 4	1 x Rs.700.00 = Rs.700.00

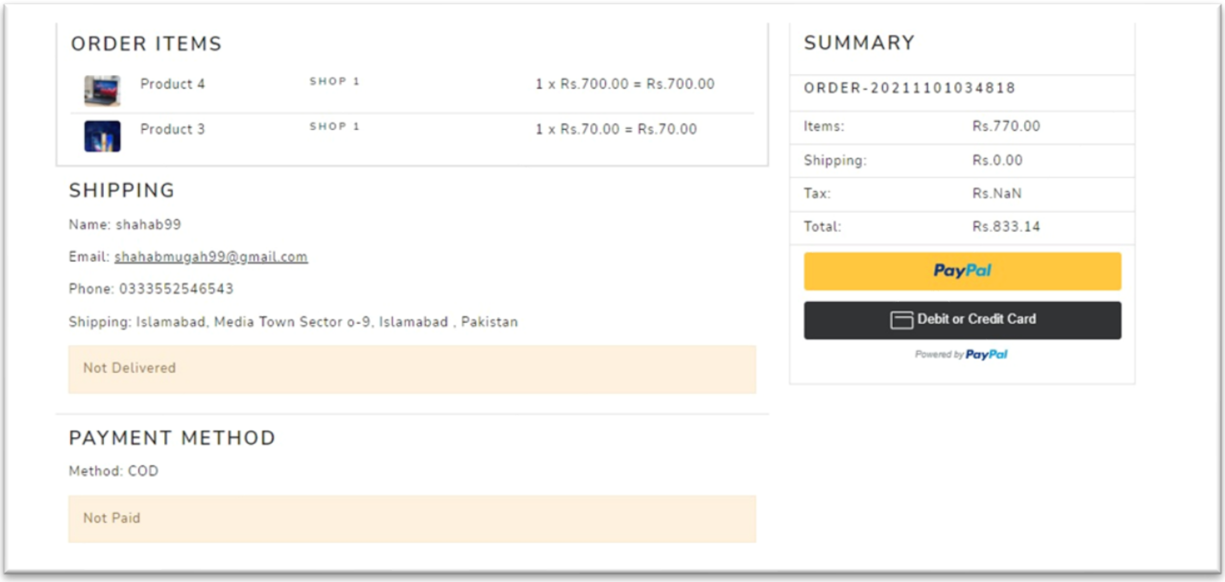
ORDER SUMMARY

Items:	Rs.770.00
Shipping:	Rs.0.00
Tax:	Rs.63.14
Total:	Rs.833.14

PLACE ORDER

Figure 4:63 Place Order Prototype

4.7.10. Order Details



ORDER ITEMS

Product 4	SHOP 1	1 x Rs.700.00 = Rs.700.00
Product 3	SHOP 1	1 x Rs.70.00 = Rs.70.00

SHIPPING

Name: shahab99

Email: shahabmugah99@gmail.com

Phone: 0333552546543

Shipping: Islamabad, Media Town Sector o-9, Islamabad , Pakistan

Not Delivered

PAYMENT METHOD

Method: COD

Not Paid

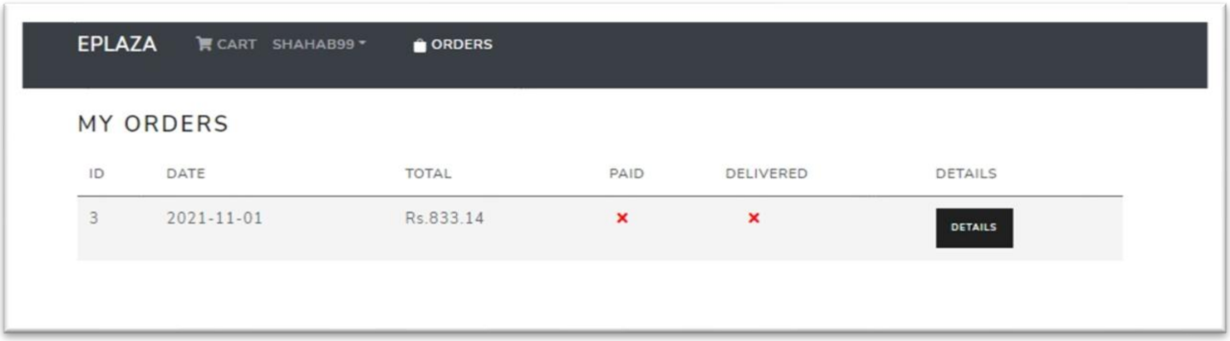
SUMMARY

ORDER-20211101034818

Items:	Rs.770.00
Shipping:	Rs.0.00
Tax:	Rs.NaN
Total:	Rs.833.14

Figure 4:64 Order Details Prototype

4.7.11. Order List

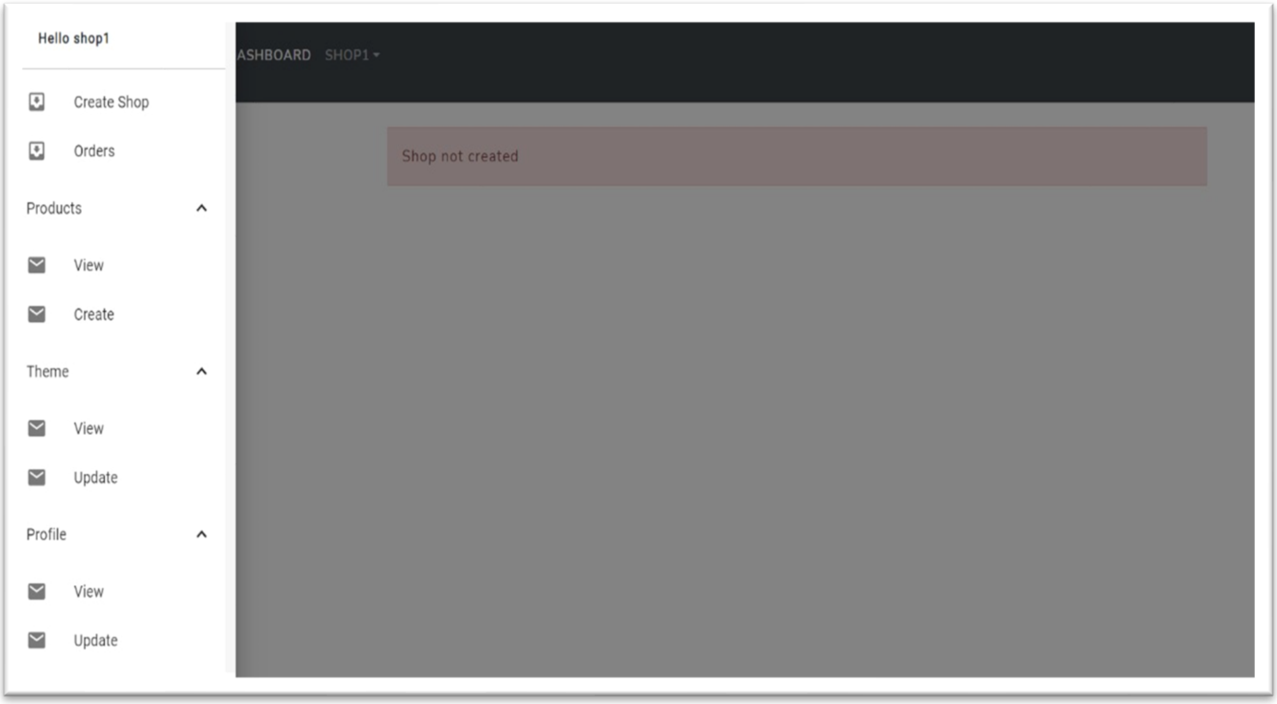


MY ORDERS

ID	DATE	TOTAL	PAID	DELIVERED	DETAILS
3	2021-11-01	Rs.833.14	✗	✗	DETAILS

Figure 4:65 Order List Prototype

4.7.12. Admin Menu



Shop not created

Figure 4:66 Seller Prototype

4.7.13. Create Product

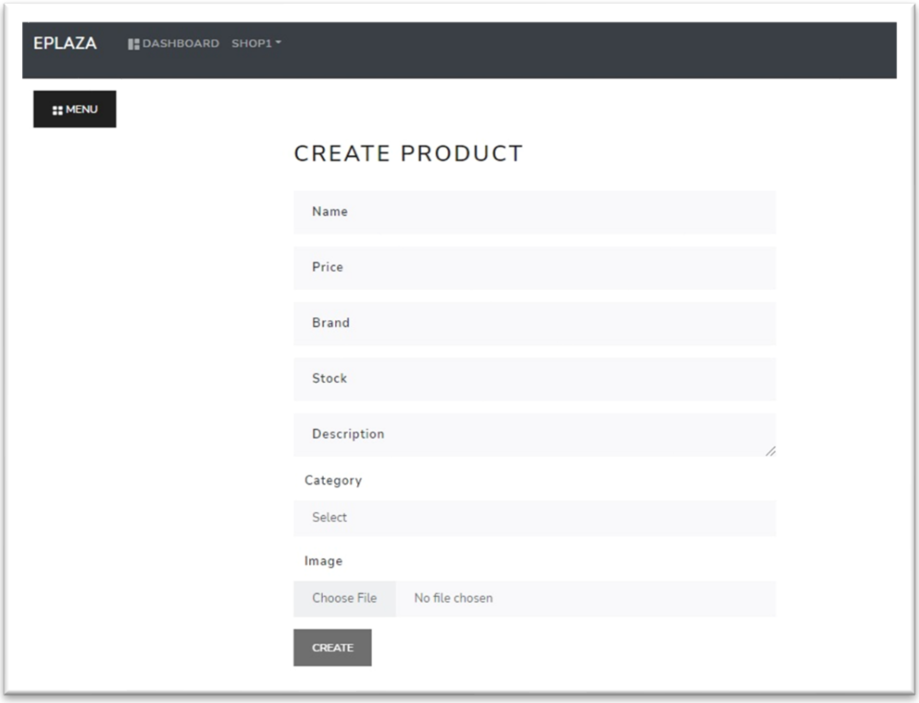


Figure 4:67 Create Product Prototype

4.7.14. Create Shop

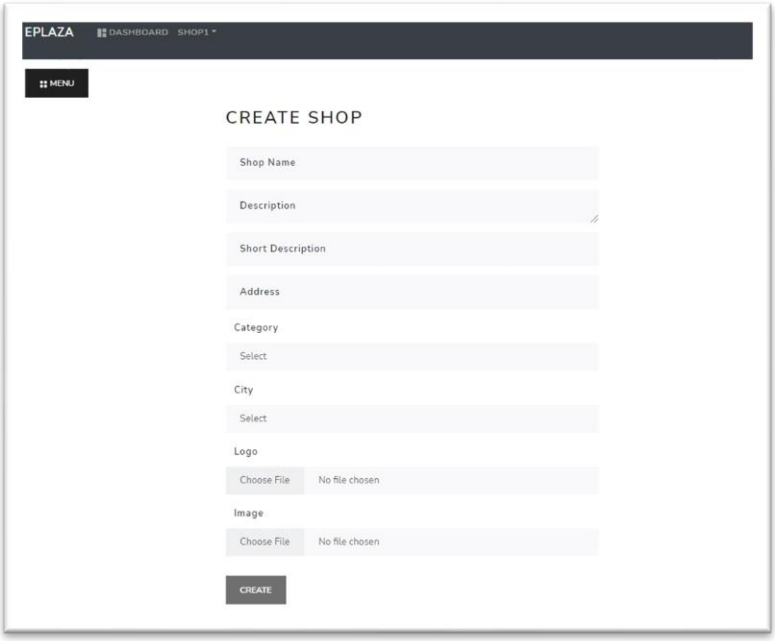


Figure 4:68 Create Shop Prototype

4.8. Conclusion

This chapter covers the system details design, structure, architecture, data models, dynamic view, detailed file structures, component representation and its Interface. By end of this chapter, we get to know internal structure, working and representation of our system.

Chapter 5

SYSTEM IMPLEMENTATION

5.1. Tools and Technologies

5.1.1. GitHub as VCS (Version Control System)

We aim to use Git, a distributed version control system that keeps track of different versions of a file (or set of files). It allowed us to access, compare, update, and share any recorded version(s) at any moment, which helped us on multiple occasions as we promoted work by creating branches.

5.1.2. PyCharm IDE (Integrated Development Environment)

PyCharm professional supported us by providing smart code completion, code inspections, on-the-fly error highlighting and quick-fixes, along with automated code refactoring and rich navigation capabilities. Not only that also helped to edit and create CSS and JS files.

5.1.3. Backend:

Django is the backend of our system, as system is mostly based on API calls so with Django Rest Framework, we are in good hands to make our system successful.

5.1.4. Frontend:

React JS is Frontend of our system, with the support of React components, Redux, Routers, use effects and use states etc. the integration of Ui is easy and stable, along with debugging is also efficient with react redux.

5.1.5. Database:

Postgres alongside Aws buckets to store images the database seems to be efficient. Mongo dB was our first choice but based on research we came across that Postgres is 15 times faster than Mongo. We also tried Aws RDS as database but due to complex pricing schemes and low efficiency we dropped it.

5.1.6. Hosting Server:

Heroku with its easy steps to deploy any project with few clicks was our first choice.

5.1.7. Architectural Pattern:

MVT (Model View Template) is the Pattern of architecture we followed as Django by default supports, it’s just like MVC. Shown in diagram 4.

5.1.8. Prototype Approach:

The prototype approach we adopted is evolutionary, the protype system tends to be final product at the end.

5.1.9. Other Authentications:

JWT (Jason Web Tokenization) is used to authenticate and encrypt the customer password, this involves tokens that expires after certain time limit making sure valid authentic user is being logged in.

5.1.10. System Libraries

Following are libraries involved in Backend and Design development.

Table 5:1 System libraries

LIBRARY	DESCRIPTION
boto3==1.18.62	For deployment.
botocore==1.21.62	For accessing Amazon web services
Django~=3.2.8	For running Django files and commands in terminal.
django-cors-headers==3.10.0	For handling server and apply security checks for API calls.
django-image-optimizer==1.0.1	Image resizing and compression.
django-storages==1.12.1	Cloud storage purpose
djangorestframework~=3.12.4	For handling rest framework libraries and commands.
djangorestframework-simplejwt==5.0.0	Web tokens for user authentication
gunicorn==20.1.0	Deployment purpose on web
Pillow~=8.3.2	Used for image handling
psycopg2==2.9.1	Handling Connection pool for Postgres
PyJWT==2.2.0	Web tokens for user authentication
s3transfer==0.5.0	For transferring data to amazon buckets
whitenoise==5.3.0	Serve static files
"axios": "^0.21.4"	For backend requests

"mdb-react-ui-kit": "^1.4.0"	For frontend design components
"react": "^17.0.2"	For handling react libraries and initiating react server
"react-bootstrap": "^2.0.0-beta.6"	For frontend components
"react-client-session": "0.0.7"	A simple object to manage client session data in a React app.
"react-dom": "^17.0.2"	This package serves as the entry point to the DOM and server renderers for React.
"react-icons": "^4.2.0"	Provide all type of icons
"react-minimal-side-navigation": "^1.9.2",	Provide side navigation bar
"react-redux": "^7.2.5",	manages state and state transformations and is often used with React
facebook-sdk==3.1.0	For Graph API to integrate Facebook page postings

5.1.11. UI Libraries

Following are frontend UI frameworks used.

5.1.12. MUI (Material User Interface)

All the major components origin belongs to the MUI this include.

- Sidebar
- Cards
- Tables
- Breadcrumbs etc.

5.1.13. React Bootstrap

UI design classes are all imported from Bootstrap.

- Columns & rows
- Paddings
- Margin
- Buttons
- Alerts
- Input fields etc.

5.1.14. Other Imports

- Email JS for emails transferring
- Framer Motion for screen transitions and hover effects
- Toastifiers for showing toast messages
- Category Tree to show categories in tree format.
- Slicker Slider for showing sliders in screen. Etc.

5.2. Development Process

We planned to divide the development of ePlaza into two main sections Base Level and Release Level. The system started from identifying functional and non-functional requirements, objective, business value and scope. Later we started establishing protype that would lead to final product.

5.2.1. Base Level

Base level is the initial phase of system covering the features shown in table 5:2. This covers the protype phase of system allowing any seller or customer to use main functionality of system.

Table 5:2 Base level features

FEATURE	DESCRIPTION
Log In	Both seller and customer can login using Django Authentications to authenticate.
Register	Both seller and customer register through Django Tokenization system
Update Profile	Both seller and customer can update his profile
Shop (CRUD)	Seller can create, update, delete, and get his shop.
Forget Password	Both customer and seller can restore password.
Product (CRUD)	Product can be seller, update, delete and read
Orders (R)	List of orders are visible to seller as well as customer
Customers Queries	Seller can respond to customer quarries
Rate/Review	Registered customer can rate and review shop or product.
Shop Selection	Customer can select his desired shop

Order Status	Order status is updated by seller for customer.
Cart updating	Item can be added, removed and quantity can be changed.
Place order	Order is placed by customer.
Create Shop	Seller can create his shop and add products to it.

5.2.2. Release Level

This phase is after the protype completion and modifying the system to cover all its functional as well as non-functional requirements. Features at release level are shown below.

Table 5:3 Release level features

FEATURE	DESCRIPTION
Discounts/Sales	New discount offers are published by seller.
Level system	Customer got opportunity to level up by placing order and based on his/her level offers are available
WhatsApp integration	Seller and customer chats through WhatsApp Sessions.
Filter product	Items can be filtered.
Email generations	Order placement and authentications mail sent.
Super Admin Restriction	The restrictions on Product entry and shop registration is to be incorporated so that everything visible to Customer is under admin (ePlaza Developers) approval

5.3. System key features

Following are the important key features making our system strong in terms of Backend as well as frontend.

5.3.1. Redux Implementation

Working with redux has been quite beneficial to us in terms of delegating state management to individual components. Instead, we design a single store that manages our states. Furthermore, the store is the primary means of communication for reading, updating, and producing data. It keeps issues like data inconsistency at bay. Furthermore, components can listen for state changes to refresh the user interface and avoid data inconsistency errors.

Finally, we can install the Redux DevTools, which provide insight into the current state of our application and make debugging and testing easier.

5.3.2. React Hooks

In our system's frontend, hooks play major role as rather than forcing a split based on lifecycle methods, hooks allow us to separate one component into smaller functions based on which elements are connected (such as setting up a subscription or requesting data). Through hooks and functional-based component the overall performance at server-side boost to 98%, which is 3% higher than a class-based component.

5.3.3. Code Splitting

React supports the bundling, which is fantastic, but as our app grows, so will our package. Especially due to huge third-party libraries are included. We must keep a watch on the code we included in our bundle to ensure that it does not become so enormous that our app takes too long to load.

It's best to get ahead of the situation and start "dividing" our bundle to avoid ending up with a large one. Splitting your app's code allows us to "lazy-load" only the features that the user requires right now, drastically improving performance. We haven't lowered the overall amount of code in your app, but we have avoided loading code that the user may never need, and we have reduced the amount of code that the user sees.

5.3.4. Object Relational Mapping (ORM)

ORMs give a high-level abstraction over a relational database, allowing a us to create, read, update, and delete data and schemas in present database using Python code rather than SQL. The option to write Python code instead of SQL helped us to speed up the development of our system, especially at the start of a project. The potential development speed improvement stems from the elimination of the need to convert from Python code to declarative SQL statements.

5.3.5. Serialization

Serialization is the process of transforming a data object (e.g., Python objects) into a format that allows us to store or transport the data, and then recreating the object using the deserialization process when needed. By transforming an object into a byte stream, serialisation allows us to send it over a network. It also aids in the preservation of the object's state. Deserialization takes less time to reconstruct an object than creating an object from a class. Serialization thus saves time.

5.3.6. Function based Views

In our system function-based views helped us in terms of easy to read, understand and implement. Making development fast and effective.

5.4. Conclusion

Above discussion related to the system development covers the main key features of system, allowing system to be faster and more effective. The imports and libraries used in our system also mentioned along with details of tools and technology that our system possesses.

Chapter 6

SYSTEM TESTING

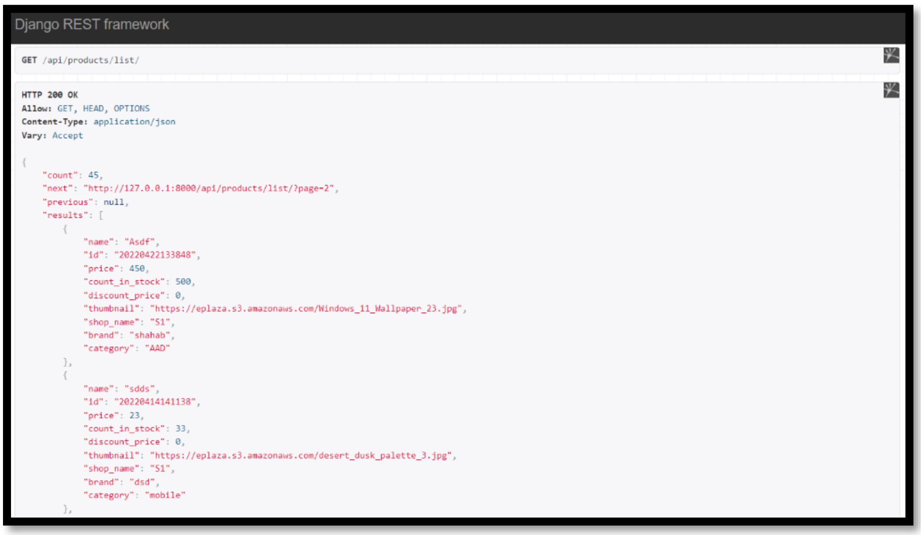
AND EVALUATION

6.1. Test Strategy

Testing is a critical component of the Software Development Life Cycle (SLDC) since it controls quality and ensures quality engineering. Before the system can be deployed, it must be tested. To evaluate the system's quality, various methodologies can be used. While developing the system, we used unit testing on each component, followed by component or module testing. We implemented a full testing framework after integrating the system by declaring specific test cases and then applying them to our system.

6.2. Unit Testing

Unit testing is a sort of testing that involves testing individual system units. Each unit is tested after it has been developed. Unit testing is used to avoid logical and business logic mistakes. Unit testing is performed during the project's development phase. Unit tests isolate a unit of code and validate its correctness. We performed unit testing using API testing with Django via CSRF tokens throughout our development.



```
GET /api/products/list/

HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "count": 45,
    "next": "http://127.0.0.1:8000/api/products/list/?page=2",
    "previous": null,
    "results": [
        {
            "name": "Asdf",
            "id": "20220422133848",
            "price": 450,
            "count_in_stock": 500,
            "discount_price": 0,
            "thumbnail": "https://eplaza.s3.amazonaws.com/Windows_11_Wallpaper_23.jpg",
            "shop_name": "S1",
            "brand": "shahab",
            "category": "AAD"
        },
        {
            "name": "sdds",
            "id": "20220414141138",
            "price": 23,
            "count_in_stock": 33,
            "discount_price": 0,
            "thumbnail": "https://eplaza.s3.amazonaws.com/desert_dusk_palette_3.jpg",
            "shop_name": "S1",
            "brand": "dsd",
            "category": "mobile"
        },
```

Figure 6:1 Unit Testing Get Product List

6.3. Component Testing

When all the system's units are completed, they are tested with one another, resulting in the creation of a component or module. This type of testing is known as component testing.

6.4. Integration

Testing Several components are brought together to finish the project during system integration. An integration test is performed when these components are combined. Individual software modules are merged and tested as a group during this step of software testing. The goal of integration testing is to determine whether a system or component complies with functional requirements.

6.5. System Testing

After the development of the entire system, the whole system is tested. It is done by applying test suits that contains pre-defined test cases for each use case.

Seller

6.5.1. Test Case for Login

Table 0:1 Seller Login

Test Scenario ID	TS-001	**Test Case ID**	TC-S01
Test Case	Login	**Test Priority**	High
Pre-Requisite	Seller Registered	**Post Requisite**	Logged In

S.No.	Inputs	Expected Output	Actual Output	Test Result	Remarks
1	Empty fields	Empty fields error popup	Same as expected	Pass	Negative Test
2	Incorrect username, password	Invalid credentials error message	Same as expected	Pass	Negative Test
3	Correct username, password	User Logged In	Same as expected	Pass	Positive Test

6.5.2. Test case Register

Table 0:2 Seller Register

Test Scenario ID		TS-002	**Test Case ID**		TC-S02
Test Case		Register	**Test Priority**		High
Pre-Requisite		Internet Connection	**Post Requisite**		Registered
S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Empty field(s)	Empty fields error popup	Same as expected	Pass	Negative Test
2	Email in invalid format	Invalid phone or email error popup	Same as expected	Pass	Negative Test
3	Username, email, phone and unmatched or weak password	Unmatched passwords or weak password error popup	Same as expected	Pass	Negative Test
4	Correct format username, password, email, phone	Username not available error popup	Same as expected	Pass	Negative Test
5	Correct format username, password, email, phone but invalid verification code	Invalid code error popup	Same as expected	Pass	Negative Test
6	Correct format credentials	Registered	Same as expected	Pass	Positive Test

6.5.3. Test Case for Email Verification

Table 0:3 Email Verification

Test Scenario ID	TS-003	**Test Case ID**	TC-S03
Test Case	Email Verification	**Test Priority**	Medium
Pre-Requisite	Working Internet Connection	**Post Requisite**	Email Verified

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Empty field	Empty fields error popup	Same as expected	Pass	Negative Test
2	Email in invalid format	Invalid email error popup	Same as expected	Pass	Negative Test
3	Correct email but invalid verification code	Invalid code error popup	Same as expected	Pass	Negative Test
4	Correct email and verification code	Email verified	Same as expected	Pass	Positive Test

6.5.4. Test Case for Update Profile

Table 0:4 Update Profile

Test Scenario ID	TS-004	**Test Case ID**	TC-S04
Test Case	Update Profile	**Test Priority**	Low
Pre-Requisite	Registered	**Post Requisite**	Profile Updated

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Empty field(s)	Empty fields error popup	Same as expected	Pass	Negative Test
2	Email or phone in invalid format	Invalid email or phone error popup	Same as expected	Pass	Negative Test
3	Unmatched password and confirm password	Passwords do not match error popup	Same as expected	Pass	Negative Test
4	Correct email, phone, and password	Profile Updated	Same as expected	Pass	Positive Test

6.5.5. Test Case for Create Shop

Table 0:5 Create Shop

Test Scenario ID	TS-005	**Test Case ID**	TC-S05
Test Case	Create Shop	**Test Priority**	High
Pre-Requisite	Registered	**Post Requisite**	Shop Created

S.No.	Inputs	Expected Output	Actual Output	Test Result	Remarks
1	Empty field(s)	Empty fields error popup	Same as expected	Pass	Negative Test
2	Valid data in all fields	Shop name already in user error message	Same as expected	Pass	Negative Test
3	Valid data in all fields	Thumbnail or logo upload error	Same as expected	Pass	Negative Test
4	Valid data in all fields	Shop created	Same as expected	Pass	Positive Test

6.5.6. Test Case for Update Shop

Table 0:6 Update Shop

Test Scenario ID	TS-006	**Test Case ID**	TC-S06
Test Case	Update Shop	**Test Priority**	Medium
Pre-Requisite	Shop exists	**Post Requisite**	Shop Updated

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Empty field(s)	Empty fields error popup	Same as expected	Pass	Negative Test
2	Valid data in all fields	Shop name already in user error message	Same as expected	Pass	Negative Test
3	Valid data in all fields	Thumbnail or logo upload error	Same as expected	Pass	Negative Test
4	Valid data in all fields	Shop updated	Same as expected	Pass	Positive Test

6.5.7. Test Case for Create Product

Table 0:7 Create Product

Test Scenario ID	TS-007	**Test Case ID**	TC-S07
Test Case	Create Product	**Test Priority**	High
Pre-Requisite	Shop exists	**Post Requisite**	Product Created

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Empty field(s)	Empty fields error popup	Same as expected	Pass	Negative Test
2	Valid data in all fields	Product name already in user error message	Same as expected	Pass	Negative Test
3	Valid data in all fields	Thumbnail upload error	Same as expected	Pass	Negative Test
4	Valid data in all fields	Product created	Same as expected	Pass	Positive Test

6.5.8. Test Case for Update Product

Table 0:8 Update Product

Test Scenario ID	TS-008	**Test Case ID**	TC-S08
Test Case	Update Product	**Test Priority**	Medium
Pre-Requisite	Product exists	**Post Requisite**	Product Updated

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Empty field(s)	Empty fields error popup	Same as expected	Pass	Negative Test
2	Valid data in all fields	Product name already in user error message	Same as expected	Pass	Negative Test
3	Valid data in all fields	Thumbnail upload error	Same as expected	Pass	Negative Test
4	Valid data in all fields	Product updated	Same as expected	Pass	Positive Test

6.5.9. Test Case for Delete Product

Table 0:9 Delete Product

Test Scenario ID		TS-009	**Test Case ID**		TC-S09
Test Case		Delete Product	**Test Priority**		Low
Pre-Requisite		Product exists	**Post Requisite**		Product Deleted
S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Select Product	Product Deleted	Same as expected	Pass	Positive Test

6.5.10. Test Case for Create Post

Table 0:10 Create Post

Test Scenario ID		TS-0010	**Test Case ID**		TC-S010
Test Case		Create Post	**Test Priority**		Low
Pre-Requisite		Logged In	**Post Requisite**		Post Created
S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Enter post content	Error accessing fb	Same as expected	Pass	Negative Test
2	Enter post content	Post created	Same as expected	Pass	Positive Test

6.5.11. Test Case for Update Post

Table 0:11 Update Post

Test Scenario ID	TS-011	**Test Case ID**	TC-S11
Test Case	Update Post	**Test Priority**	Low
Pre-Requisite	Post exists	**Post Requisite**	Post Updated

S.No.	Inputs	Expected Output	Actual Output	Test Result	Remarks
1	Enter post id and content	Post not found error	Same as expected	Pass	Negative Test
2	Enter post id and content	Error accessing fb	Same as expected	Pass	Negative Test
3	Enter post id and content	Post created	Same as expected	Pass	Positive Test

6.5.12. Test Case for Create Coupon

Table 0:12 Create Coupon

Test Scenario ID	TS-012	**Test Case ID**	TC-S12
Test Case	Create Coupon	**Test Priority**	Medium
Pre-Requisite	Logged In	**Post Requisite**	Coupon Created

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Empty field(s)	Empty fields error popup	Same as expected	Pass	Negative Test
2	Valid data in all fields	Coupon code already used error	Same as expected	Pass	Negative Test
3	Valid data in all fields	Coupon created	Same as expected	Pass	Positive Test

6.5.13. Test Case for Update Coupon

Table 0:13 Update Coupon

Test Scenario ID	TS-013	**Test Case ID**	TC-S13
Test Case	Update Coupon	**Test Priority**	Medium
Pre-Requisite	Coupon exists	**Post Requisite**	Coupon Updated

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Empty field(s)	Empty fields error popup	Same as expected	Pass	Negative Test
2	Valid data in all fields	Coupon code already used error	Same as expected	Pass	Negative Test
3	Valid data in all fields	Coupon updated	Same as expected	Pass	Positive Test

6.5.14. Test Case for Delete Coupon

Table 0:14 Delete Coupon

Test Scenario ID	TS-014	**Test Case ID**	TC-S14
Test Case	Delete Coupon	**Test Priority**	Medium
Pre-Requisite	Coupon exists	**Post Requisite**	Coupon Deleted

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Select coupon	Coupon deleted	Same as expected	Pass	Positive Test

6.5.15. Test Case for Update Product Discount

Table 0:15 Update Product Discount

Test Scenario ID	TS-015	**Test Case ID**	TC-S15
Test Case	Update Product Discount	**Test Priority**	Medium
Pre-Requisite	Product exists	**Post Requisite**	Product Discount Updated

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Select Product and enter amount	Amount greater than 80% of price error popup	Same as expected	Pass	Negative Test
2	Select Product and enter amount	Discount updated	Same as expected	Pass	Positive Test

6.5.16. Test Case for Update Order Status

Table 0:16 Update Order Status

<table>
<tr><td colspan="2"><b>Test Scenario ID</b></td><td>TS-016</td><td colspan="2"><b>Test Case ID</b></td><td>TC-S16</td></tr>
<tr><td colspan="2"><b>Test Case</b></td><td>Update Order Status</td><td colspan="2"><b>Test Priority</b></td><td>High</td></tr>
<tr><td colspan="2"><b>Pre-Requisite</b></td><td>Order exists</td><td colspan="2"><b>Post Requisite</b></td><td>Order Status Updated</td></tr>
<tr><td><b>S.No.</b></td><td><b>Inputs</b></td><td><b>Expected Output</b></td><td><b>Actual Output</b></td><td><b>Test Result</b></td><td><b>Remarks</b></td></tr>
<tr><td>1</td><td>Select Order and select status</td><td>Error popup</td><td>Same as expected</td><td>Pass</td><td>Negative Test</td></tr>
<tr><td>2</td><td>Select Order and select status</td><td>Status updated</td><td>Same as expected</td><td>Pass</td><td>Positive Test</td></tr>
</table>

Customer

6.5.17. Test Case for Customer Login

Table 0:17 Customer Login

Test Scenario ID	TS-017	**Test Case ID**	TC-S17
Test Case	Login	**Test Priority**	High
Pre-Requisite	Customer Registered	**Post Requisite**	Logged In

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Empty fields	Empty fields error popup	Same as expected	Pass	Negative Test
2	Incorrect username, password	Invalid credentials error message	Same as expected	Pass	Negative Test
3	Correct username, password	User Logged In	Same as expected	Pass	Positive Test

6.5.18. Test Case for Customer Register

Table 0:18 Customer Register

Test Scenario ID	TS-018	**Test Case ID**	TC-S18
Test Case	Register	**Test Priority**	High
Pre-Requisite	Working Internet Connection	**Post Requisite**	Registered

S.No.	Inputs	Expected Output	Actual Output	Test Result	Remarks
1	Empty field(s)	Empty fields error popup	Same as expected	Pass	Negative Test
2	Username, pass and phone or email in invalid format	Invalid phone or email error popup	Same as expected	Pass	Negative Test
3	Username, email, phone and unmatched or weak password	Unmatched passwords or weak password error popup	Same as expected	Pass	Negative Test
4	Correct format username, password, email, phone	Username not available error popup	Same as expected	Pass	Negative Test
5	Correct format username, password, email, and phone	Registered	Same as expected	Pass	Positive Test

6.5.19. Test Case for Update Profile

Table 0:19 Update Profile

Test Scenario ID	TS-019	**Test Case ID**	TC-S19
Test Case	Update Profile	**Test Priority**	Low
Pre-Requisite	Registered	**Post Requisite**	Profile Updated

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Empty field(s)	Empty fields error popup	Same as expected	Pass	Negative Test
2	Email or phone in invalid format	Invalid email or phone error popup	Same as expected	Pass	Negative Test
3	Unmatched password and confirm password	Passwords do not match error popup	Same as expected	Pass	Negative Test
4	Correct email and phone, city, address, and password	Profile Updated	Same as expected	Pass	Positive Test

6.5.20. Test Case for Recover Password

Table 0:20 Recover Password

Test Scenario ID	TS-020	**Test Case ID**	TC-S20
Test Case	Recover Password	**Test Priority**	Medium
Pre-Requisite	Registered	**Post Requisite**	Password Changed

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Empty field	Empty fields error popup	Same as expected	Pass	Negative Test
2	Invalid username	username not found error popup	Same as expected	Pass	Negative Test
3	Correct username	new pass sent on mail popup	Same as expected	Pass	Positive Test

6.5.21. Test Case for Search Product

Table 0:21 Search Product

Test Scenario ID	TS-021	**Test Case ID**	TC-S21
Test Case	Search Product	**Test Priority**	Medium
Pre-Requisite	Internet Working	**Post Requisite**	Product(s) Displayed

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Empty field	Button click disabled	Same as expected	Pass	Negative Test
2	Enter keyword	No matching product(s) found	Same as expected	Pass	Negative Test
3	Enter keyword	Matching product(s) displayed	Same as expected	Pass	Positive Test

6.5.22. Test Case for Filter Product

Table 0:22 Filter Product

Test Scenario ID	TS-022	**Test Case ID**	TC-S22
Test Case	Filter Product	**Test Priority**	Medium
Pre-Requisite	Internet Working	**Post Requisite**	Product(s) Displayed

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	No filter selected	Button click disabled	Same as expected	Pass	Negative Test
2	Filters selected	No matching product(s) found	Same as expected	Pass	Negative Test
3	Filters selected	Matching product(s) displayed	Same as expected	Pass	Positive Test

6.5.23. Test Case for Add to Cart

Table 0:23 Add to Cart

Test Scenario ID	TS-023	**Test Case ID**	TC-S23
Test Case	Add To Cart	**Test Priority**	High
Pre-Requisite	Logged In	**Post Requisite**	Added

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Product out of stock	Button click disabled	Same as expected	Pass	Negative Test
2	Selected quantity zero	Button click disabled	Same as expected	Pass	Negative Test
3	Selected quantity greater than zero	Added to cart	Same as expected	Pass	Positive Test

6.5.24. Test Case for Delete from Cart

Table 0:24 Delete from Cart

Test Scenario ID	TS-024	**Test Case ID**	TC-S24
Test Case	Delete From Cart	**Test Priority**	High
Pre-Requisite	Product in Cart	**Post Requisite**	Deleted

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Select Product	Removed from cart	Same as expected	Pass	Positive Test

6.5.25. Test Case for Place Order

Table 0:25 Place Order

Test Scenario ID	TS-025	**Test Case ID**	TC-S25
Test Case	Place Order	**Test Priority**	High
Pre-Requisite	Product(s) in cart	**Post Requisite**	Order Placed

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Checkout, Select Payment Method, Add Shipping Info	Error Message Displayed	Same as expected	Pass	Negative Test
2	Checkout, Select Payment Method, Add Shipping Info	Mail not sent error message	Same as expected	Pass	Negative Test
3	Checkout, Select Payment Method, Add Shipping Info	Customer Level not updated error message	Same as expected	Pass	Negative Test
4	Checkout, Select Payment Method, Add Shipping Info	Order Placed	Same as expected	Pass	Positive Test

6.5.26. Test Case for Track Order

Table 0:26 Track Order

Test Scenario ID	TS-026	**Test Case ID**	TC-S26
Test Case	Track Order	**Test Priority**	Medium
Pre-Requisite	Order Placed	**Post Requisite**	Order Tracked

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Enter Order Id	Order not found error message displayed	Same as expected	Pass	Negative Test
2	Enter Order Id	Order Tracked Details shown	Same as expected	Pass	Positive Test

6.5.27. Test Case for Review Product

Table 0:27 Review Product

Test Scenario ID	TS-027	**Test Case ID**	TC-S27
Test Case	Review Product	**Test Priority**	Medium
Pre-Requisite	Product Purchased	**Post Requisite**	Product Reviewed

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Rate Product and Enter comment	Not logged in error message displayed	Same as expected	Pass	Negative Test
2	Rate Product and Enter comments	Product not purchased error message displayed	Same as expected	Pass	Negative Test
3	Rate Product and Enter comments	Product already reviewed message displayed	Same as expected	Pass	Negative Test
4	Enter Order Id	Product Reviewed	Same as expected	Pass	Positive Test

Admin

6.5.28. Test Case for Login

Table 0:28 Admin Login

<table>
<tr><td colspan="2"><b>Test Scenario ID</b></td><td colspan="2">TS-028</td><td><b>Test Case ID</b></td><td>TC-S28</td></tr>
<tr><td colspan="2"><b>Test Case</b></td><td colspan="2">Login</td><td><b>Test Priority</b></td><td>High</td></tr>
<tr><td colspan="2"><b>Pre-Requisite</b></td><td colspan="2">Account exists</td><td><b>Post Requisite</b></td><td>Logged In</td></tr>
<tr><td><b>S.No.</b></td><td><b>Inputs</b></td><td><b>Expected Output</b></td><td><b>Actual Output</b></td><td><b>Test Result</b></td><td><b>Remarks</b></td></tr>
<tr><td>1</td><td>Empty fields</td><td>Empty fields error popup</td><td>Same as expected</td><td>Pass</td><td>Negative Test</td></tr>
<tr><td>2</td><td>Incorrect username, password</td><td>Invalid credentials error message</td><td>Same as expected</td><td>Pass</td><td>Negative Test</td></tr>
<tr><td>3</td><td>Correct username, password</td><td>Admin Logged In</td><td>Same as expected</td><td>Pass</td><td>Positive Test</td></tr>
</table>

6.5.29. Test Case for Verify Shop

Table 0:29 Verify Shop

Test Scenario ID	TS-029	**Test Case ID**	TC-S29
Test Case	Verify Shop	**Test Priority**	High
Pre-Requisite	Shop created	**Post Requisite**	Shop Verified

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Select Shop	Shop verified	Same as expected	Pass	Positive Test

6.5.30. Test Case for Delete User

Table 0:30 Delete User

Test Scenario ID	TS-030	**Test Case ID**	TC-S30
Test Case	Delete User	**Test Priority**	High
Pre-Requisite	User exists	**Post Requisite**	User Deleted

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Select User	User Deleted	Same as expected	Pass	Positive Test

6.5.31. Test Case for Update Utils

Table 0:31 Update Utils

Test Scenario ID	TS-031	**Test Case ID**	TC-S31
Test Case	Update Utils	**Test Priority**	High
Pre-Requisite	Logged In	**Post Requisite**	Utils Updated

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Empty field(s)	Empty fields error popup	Same as expected	Pass	Negative Test
2	Valid Data	Utils Updated	Same as expected	Pass	Positive Test

6.5.32. Test Case for Update User Details

Table 0:32 Update User Details

Test Scenario ID	TS-032	**Test Case ID**	TC-S32
Test Case	Update User Details	**Test Priority**	High
Pre-Requisite	User exists	**Post Requisite**	Details Updated

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Empty field(s)	Empty fields error popup	Same as expected	Pass	Negative Test
2	Email or phone in invalid format	Invalid email or phone error popup	Same as expected	Pass	Negative Test
3	Unmatched password and confirm password	Passwords do not match error popup	Same as expected	Pass	Negative Test
4	Correct email, phone, and password	Profile Updated	Same as expected	Pass	Positive Test

6.5.33. Test Case for Update Order Status

Table 0:33 Update Order Status

Test Scenario ID	TS-033	**Test Case ID**	TC-S33
Test Case	Update Order Status	**Test Priority**	High
Pre-Requisite	Order exists	**Post Requisite**	Status Updated

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Select Order and select status	Error popup	Same as expected	Pass	Negative Test
2	Select Order and select status	Status updated	Same as expected	Pass	Positive Test

6.5.34. Test Case for Add Category

Table 0:34 Add Category

Test Scenario ID	TS-034	**Test Case ID**	TC-S34
Test Case	Add Category	**Test Priority**	High
Pre-Requisite	Logged In	**Post Requisite**	Category Added

S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Enter name	Category already exists	Same as expected	Pass	Negative Test
2	Enter name	Category added	Same as expected	Pass	Positive Test

6.5.35. Test Case for Add Carousal Category

Table 0:35 Add Carousal Category

Test Scenario ID		TS-035	**Test Case ID**		TC-S35
Test Case		Add Carousal Category	**Test Priority**		High
Pre-Requisite		Category exists	**Post Requisite**		Category Added
S.No.	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Remarks**
1	Select Category	Category already exists	Same as expected	Pass	Negative Test
2	Select Category	Category added	Same as expected	Pass	Positive Test

6.6. Alpha Testing

Alpha testing is the first step in determining whether ePlaza functions as intended. These tests are conducted early in the development process by us to determine the system functionality based on SRS and identifying issues that can cause problem to end users. Following are some issues encountered.

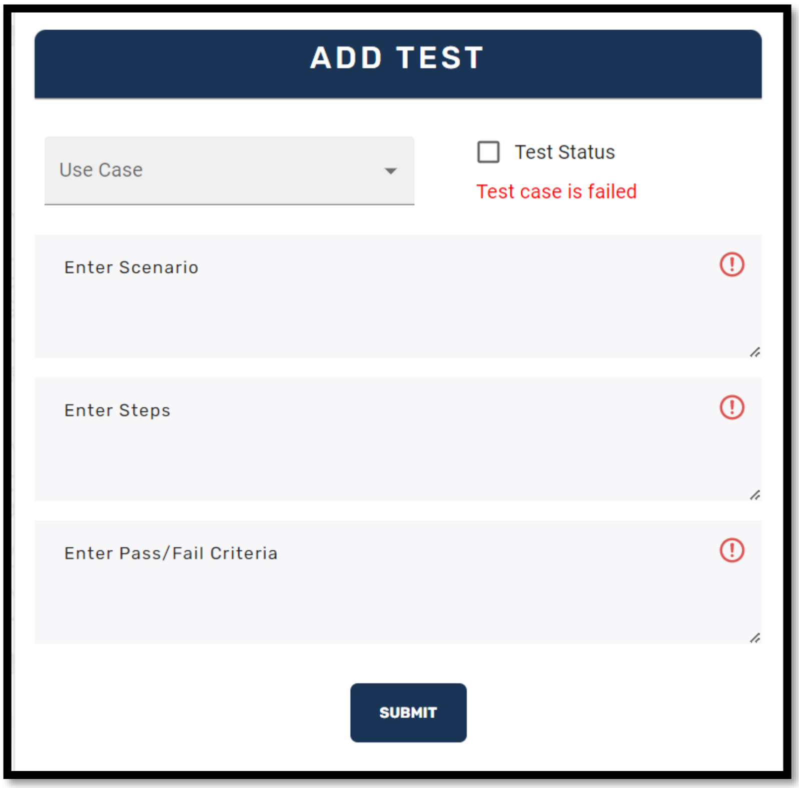
- Product filtration not working
- Email not receiving
- Order placement causing problem in models
- Cart not updating.

6.7. Beta Testing

Beta testing was carried out by making the system online and handing it over to our friends and family members, allowing them to test the system capabilities and provide comments. These were.

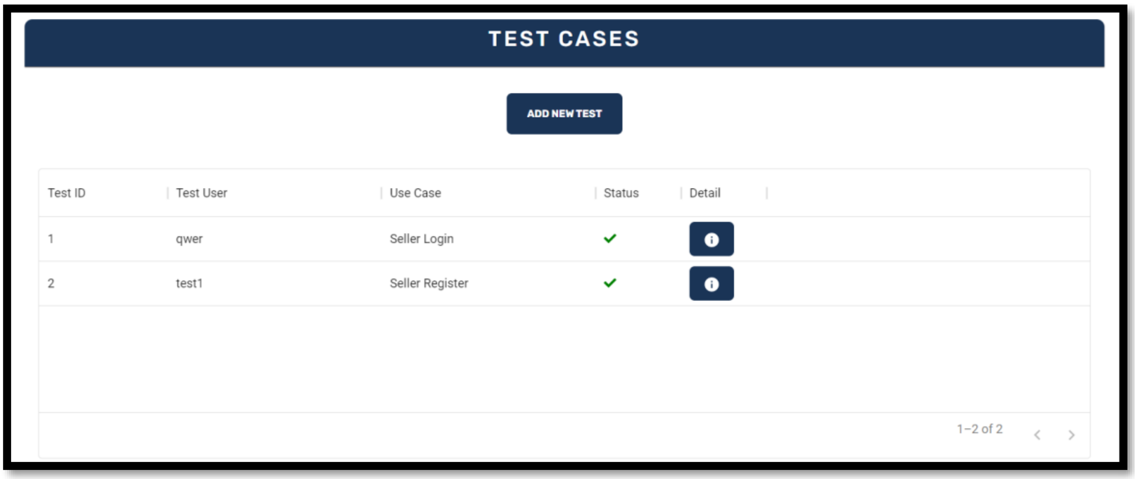
- Customer email validation checks are causing complications.
- Broken and dead links were discovered.
- Errors in cart updating
- Order placement is delayed.
- UI interaction ease not present

Aside from that, we established a testing module that is integrated with our system to verify the pass and fail criteria for all use cases. The testing module is being handed over to our university fellows, who can login, enter test details, test the system against any use case and mark then it as pass or fail.



Test case is failed

Figure 6:2 Testing Home Screen



TEST CASES

ADD NEW TEST

Test ID	Test User	Use Case	Status	Detail
1	qwer	Seller Login	✓	ⓘ
2	test1	Seller Register	✓	ⓘ

1–2 of 2

Figure 6:3 Test Report

6.8. Conclusion

The system is tested from various angles at the end of the chapter utilising Test suits. However, by making the beta version of the system available to the public, testing and assessment of the system is made more efficient. With the help of alpha and beta testing, we can ensure that ePlaza can perform well in the market.

Chapter 7

CONCLUSION

7.1 Contributions

The document addresses both technological and non-technical limitations. Based on our goals, the system tends to benefit both local sellers and customers. The system has appealing elements to entice clients to visit ePlaza and allows social sellers to combine their operations with ours.

7.2 Reflections

Our comments on the system are based on how easy the system allows the seller to register store, publish desired products, and manage orders. In terms of customers, how frequently are they eager to visit our site and use our platform? Developmental ideas can be explained in terms of their strengths and flaws.

7.2.1 System Strength

Our system's strength resides in its architecture, file structure, schemas, tools usage, and conventions adhered to. We intend to lay a solid basis for the overall architecture of our system. Our system includes proper authentication, API or system calls, and simple debugging.

7.2.2 System Shortcomings

Despite its strong structure, the system has shortcomings based on the type of functionalities we provide, such as a search engine that is ineffective because no search algorithms have been established, filtration that is still not up to par with other ecommerce sites, and despite tough competition in the market, our ecommerce platform only covers functionalities within its domain.

7.3 Future Work

ePlaza's future development will rely on it refining the user interface, adding new capabilities and schemas, and integrating machine learning or AI to the system. According to the development strategy, new technologies will be integrated alongside the enhancement of the search engine and filtration system.

- Bargaining Chat Bots
- Engine for advanced suggestions.
- Shop customizations.
- Admin and seller stats improvements.

We also intend to expand ePlaza through Android and iOS in the future to capture more market.

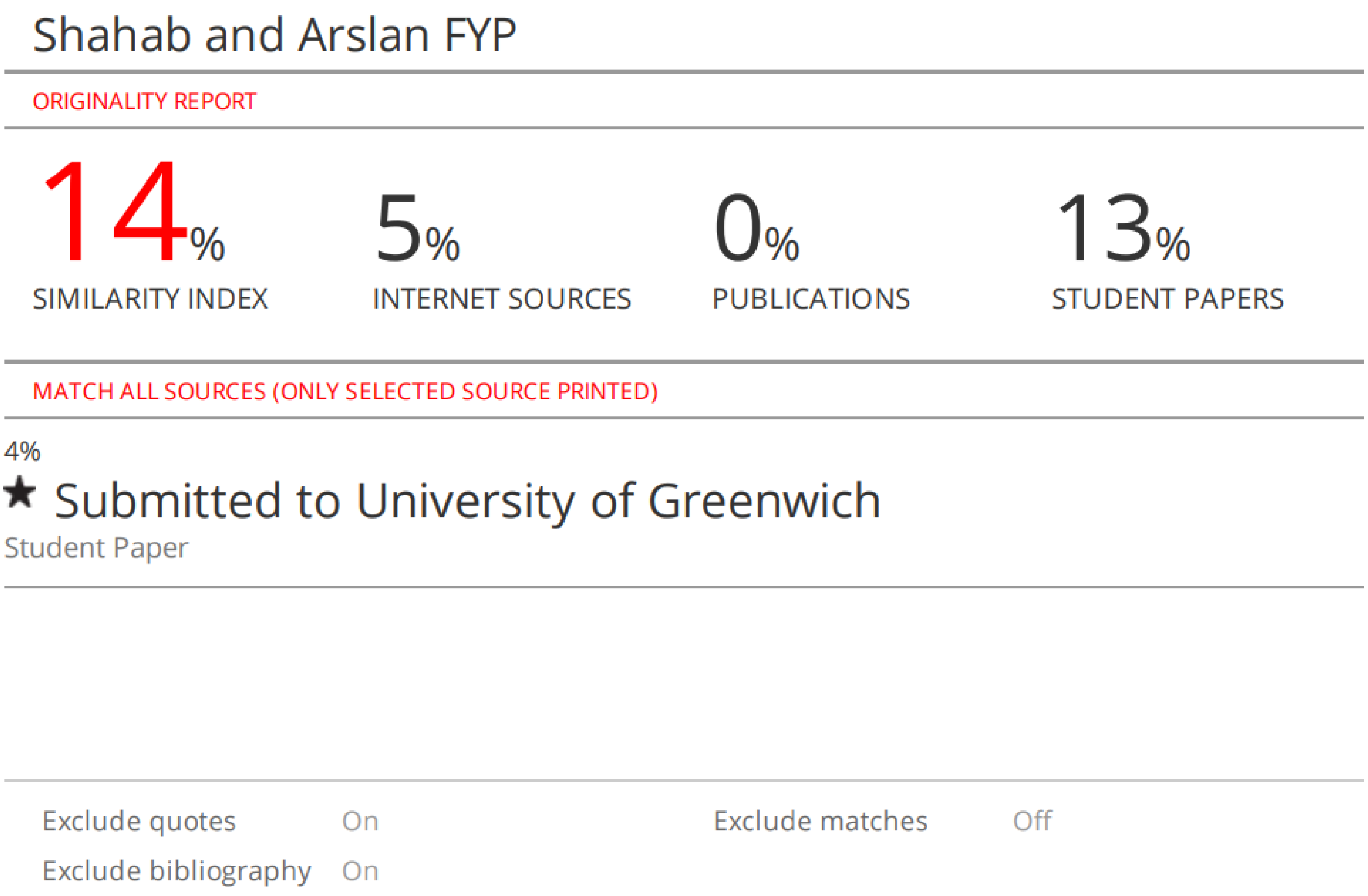
7.4 References

[1] Hongyan Xin School of Economic Management, Beihua University, Jilin City, China.” Summary of e-Commerce Research Move”.*Research on E-Commerce Business Model Based on Intelligent Terminals Mobile.* *https://ieeexplore.ieee.org/document/7429209/*

[2] Noorfadzilah Md Zainudin Department of Computer & Information Sciences, Universiti Teknologi PETRONAS, Sri Iskandar, Perak, Malaysia.” Results and Discussion”. *Designing e-commerce user interface.* https://ieeexplore.ieee.org/document/5716744/

[3] Syed Emdad Ullah Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh, Tania Alauddin Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh,Hasan U. Zaman Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh.” SECTION IV.Development Tools ”. *Developing an E-commerce website* .https://ieeexplore.ieee.org/document/7522526/

[4] Jianchi Xiang Sch. of Bus., Central South Univ., Changsha.” SECTION III.Website Customer Satisfaction Model”. *Customer Satisfaction of E-Commerce Websites*. https://ieeexplore.ieee.org/document/5072797/.



Shahab and Arslan FYP

ORIGINALITY REPORT

14%	5%	0%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

4%

★ Submitted to University of Greenwich

Student Paper

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		